Automating the Dynamic Balancing of a Multiplayer Competitive Game Using Data Aggregated from each Play Session

Batu Aytemiz, Jay Chen
Department of Computer Science, New York University Abu Dhabi
Saadiyat Island, Abu Dhabi, United Arab Emirates
batu.aytemiz@nyu.edu

Abstract

One of the most important factors when it comes to making a multiplayer competitive game is the balance of the system. Currently the industry standard in balancing competitive multiplayer games is aggregating data from thousands, millions in certain cases, of game sessions analysing it and manually coming up with an update every three or four months. We attack this problem head on by trying to automate this process. The paper proposes the development of a competitive multiplayer game designed specifically to have an build in automated balance system that dynamically alters the balance of the game. We try to show the effects of such an system and try to understand whether automating the balance process increases the overall enjoyability of the game and/or make the game objectively more balanced.

1. Introduction

When we look at the current gaming landscape we can see that E-sports and competitive games are becoming incredibly popular. In 2015 The International, the biggest DOTA 2 tournament, has reached over 18 million dollars [4] in its prize pool. Another extremely popular game, League of Legends, has over 32 million monthly active players each month reported[5] by Riot Games. The enjoyment people get from multiplayer competitive games like these, and many others, is mainly based on the balance of the game. Players want to play games where many different strategies are viable and there is no single degenerate style of competing.

Most competitive games give players different choices in how they want to play the game. This can mean being able to choose different equipments, characters or abilities. In an ideal scenario all these options should be viable and be balanced power-wise. However currently this task is nigh on impossible accomplish perfectly and this results in the player base focusing on a handful of strategies (this can come in forms of choosing certain weapons or characters to play) instead of utilizing everything the game has to offer. (Here

you can see the play rates of different heroes in a game called DOTA 2.)

Currently the go to solution that the industry is applying in most cases is for the developers of the game to patch and update their product every 3-4 months, tweaking the different values (attack, speed, armor, health, etc…) they have in their game strengthening weaker strategies and weakening the most popular ones. This process is slow and is success is dependent on the developer's discretion.

In their paper Hunicke and Chapman state that game balancing can not be automated. [3] However I still believe that this concept is worth the attempt. I am proposing building a system that automates the balancing of the given competitive game. Instead of changing the values every few months, the game auto corrects the balance dynamically as the players play the game based on the user data that is being generated. Each time a game session we can use the information to answer many questions: How successful were the players using tool X? What is the victory rate of tool Y vs Z? How many people decide to play with the tool T? etc… I believe feeding these kinds of questions into a program can allow the it to make the necessary changes to ensure that the player experience is always as positive as it could be. I foresee that there would be many implications of such a system:

- The game will never have a stale metagame. Metagame refers to the decisions players make before the game starts such as creating their teams, choosing their weapons and coming up with strategies. Not having a stale metagame means the tools that are used the most (or have the highest win percentages) will be constantly changing. This is incredibly important for a healthy gaming ecosystem. The over usage of the identical tools, and thus having a stale metagame, makes the game less exciting for the players and reduce the overall enjoyment of the game.

- Players will be encouraged to constantly come up with new ideas and plans, rewarding out of the box thinking. One of the most rewarding feelings in these kinds of competitive games is being able create new tactics. The ever changing power levels will allow tools that were underutilized to be competitive once again and there will always be a race to discover these new opportunities. This tension will keep the game fresh.

- If we were to push the limits of our system the meta-game can be susceptible for organized changes by big groups. (unlikely but possible) This can introduce a whole new layer of strategizing when it comes to competitive aspects of the games.

2. Game Balancing Techniques

There are several different types of balancing styles that game developers utilize. In the most general sense developers try to balance the game based on the difficulty of the system. A perfectly balanced game is one that matches the skill level of the player. However this type of balancing can only be utilized in single player games. In multiplayer games as there are more than one player that is participating in any given session establishing a balance by using the skill level of one player would mean that for every other player the game would feel unbalanced. Instead what we can do is to make sure that there is no power difference between the possible tools that the players can use. This separation however does not mean we cannot learn from the balancing systems that single player games use and implement them in one way or another to enhance the final design.

There are several styles of difficulty adjustment styles, with four of them being the more prominent ones: [1] Matchmaking, difficulty adjustment, asymmetric roles and skill assistance.

2.1. Matchmaking

Matchmaking refers to the system in which in any given play session players of similar caliber are matched with each other. [7] This system is utilized in most if not all of multiplayer competitive games to make sure each session is as fair as it gets. In addition to trying to match people of the same skill level matchmaking systems also try to minimize external effects that would affect the game experience such as lag and internet connection speed.

This system will be implemented into the proposed system as our metric of success is whether the game itself and the tools it provides to the player is balanced against each other or not. This type of analysis can be made most accurately only when the player's skill level are as close to each other as possible.

2.2. Asymmetric Roles

This refers to having a selection of roles in any given play session that the players can chose from; some more difficult to play than others. This allows players of different caliber to contribute to the team in similar amounts by dividing the roles that require different levels of skill accordingly.

A similar style will be implemented in the proposed game to allow players different play styles. Even though the proposed game is not a team based game one of the main goals is to provide an experience where different tools are all viable in a competitive aspect.

2.3. Difficulty Adjustment

This type of balancing refers to the player consciously changing different parameters to adjust the difficulty of the game to their skill level. The static approach style enables the players to choose the difficulty they want to play in before the game starts. [1] This

approach has varied problems. First the player has not yet experienced the game and is making a decision based on assumptions. Second the description of the difficulty levels are usually not quantitative to offer enough information. Third the perceived game difficulty is completely subjective as it is a function of the relationship between the player skill and the obstacles players face in the game, not an absolute scale of the latter. This style also fails to solve another problem. Human players enhance their skills as they are playing the game[11] which results in the initial static decision to obsolete regardless of the initial accuracy.

However a smarter way of implementing difficulty adjustment is embedding it as a mechanic with in the game. Bastion by Supergiant games does exactly that. In Bastion the players do not select a difficulty level to start the game but instead they are given the option to [8] accept different challenges that adjust the difficulty dynamically. Activating these challenges also gives the player an experience and money boost. This creates a tension on the player to always push for the limits. This particular style works especially well for several reasons. First of all there is an in game reason for these challenges which fits well with the narrative they created. In Bastion the player activates these challenges by praying to given idols. The second reason is that not every challenge is the same and thus players can pick and choose different ways to make the game more difficult. Some of the examples are as follows: Enemy movement and attack speed is increased, enemies drop a bomb when defeated, enemies randomly turn translucent which causes all of your attacks to pass through them etc… As for a multiplayer game however player controlled difficulty is not optimal.

2.4 Dynamic Difficulty Adjustment

Dynamic Difficulty Adjustment (DDA) system offers an alternative to the static difficulty adjustment we discussed above by modulating the in-game systems in order to react to the particular player's skill over the course of the single game session. [2]

There are many games that implement this system in single player games or even cooperative multiplayer games like Left 4 Dead and Left 4 Dead 2 by Valve.[6] In these games an AI director is constantly gathering data from the actions of the players. Based on the information gathered it dynamically adjusts the difficulty of the stage. It looks at different perimeters such as: The total health of the characters, the total ammunition of the characters, total amount of equipment carried by the players, distance traveled from the safe point, proximity to the objectives, the accuracy of the players etc… All of these separate variables change and contribute to a general "stress level" of the players. If the players have high amount of health points but they are close the the target the AI director responds by spawning more enemies. If the players are struggling the Director gives them

breathing room by lowering the zombies that have been spawned.

We will be adopting a part of Dynamic Difficulty Adjustment system. Even though we will be tracking similar data from the player's behavior the game does not use DDA for changing the immediate play session. Instead the collected information would be used to alter the next play session, not only for the given players but globally.

2.5. Restricted Play

Another more experimental tool used by game designer to balance a competitive game is called Restricted Play. This framework tries to identify what moves that a player can take in order to increase the odds of success. The way to test whether one tool is more powerful than the others is by measuring the win rate of an of a player who is restricted from ever (or often) using that tool, against a player who exploits this restriction.[9]

Even though this technique can provide some insight to the different balance level of the proposed game it falls short in one criteria to be considered applicable in our design. It is not fun. In no event restricting one player's tools is considered fun against a fully capable opponent. This method is useful for moderated and specialized play tests however in the proposed game the data will be collected from standard game sessions of the users. For this reason explicitly hampering one party goes against the main philosophy of this design.

2.6. Skill Assistance

This style refers to giving mechanical help dynamically to the underperforming players. The game proposed will not utilize this technique as it aims to create the most even platform where the only reasons that players perform differently is the skill of both parties and not any other artifact of game design or balance decisions. It is also shown that whenever the better player realized his opponent is getting mechanical help he can use this knowledge to adapt their tactics and exploit the situation. [10]

3. Implementation of the Game System

After surveying the possible options I have decided that no game that I have access to would allow enough altering in the source code to test my implementation of a self-balancing multiplayer game. For this reason I decided to design and develop a game to be the testing grounds for my proposed system. In order to better achieve my goal I have decided to set some parameters and objectives to guide me in my design and development process.

- The game has to be **fun**. If people do not play the game there won't be enough data to make the necessary changes to the game balance. Making a game that people want to play is of the utmost importance. The man

hours I can voluntarily gather the better as there will be a difference between those who play due to internal motivation vs. those who play the game due to external motivation.

- The game has to be **simple to learn**, hard to master. Each play session should also be relatively short. This will allow people to test my game and generate valid data in their first play session. If the learning curve is too steep I will have to retain the same users for my experiments. In the ideal scenario the game would establish a player base that would react to the changing balance, however it is important to design the game so that in the event of the game not reaching critical mass we should still be able to extract usable information from new players coming in.

- The game has to be **good looking** while being **visually minimalistic**. The focus of the game will purely be on the gameplay aspects and the mechanical depth as it is first and foremost a competitive game. I believe allocating my resources in other places would serve me well going forward.

- The game has to be able to offer very **distinct play styles** so that these different styles can be balanced against each other. In essence we are trying to create a game that offers a variety of different but all viable tools to the player. If there are not enough different tools for the players to choose from the balance readings will not be reflective of a real life scenario.

- Have many **tuning knobs** for the algorithms to affect the balance of the game. Without play testing I will not know for certain which values will be tweaked by the automated balancer. In order to find values that are impactful enough to make a difference but not powerful enough to totally swing the balance upside down I need many tuning knobs. At this given moment here are the possible options: Health, Attack Damage, Attack Speed, Range, Reload Speed, Ammo Count, Movement Speed.

- Have a **balance** between **skill** based gameplay and **number** based gameplay. (Skill based gameplay refers to the games in which your character is identical and what differentiates you from other players is your skill and numbers based gameplay refers to game where your characters values differentiates you from other players.)

These values in mind I have looked at possible inspiration points to decide what kind of a game I would be designing. Here are a important ones that shaped the final design in one way or another.

Killer Queen[12]

This is a 5 on 5 game that gives the players different ways to win the match. Initially this type of game seemed to be a good option as it inherently had asymmetric roles built into it. I would have balanced the different objectives that the player had to accomplish in order to winning. However the problem with this is that as both sides have access to all roles at any given time balancing would just empower some roles over the others but not affect the larger balance at hand. Another problem is that this type of games are best played in teams. From a logistics perspective it would be much easier for me to test a one on one game and from a technical perspective it would better emphasize the changes minimizing the effect of skill.

Rocket League[13] / Sportfriends[14]

Both Rocket League and Sportfriends as a competitive sports game. In rocket league you play soccer with remote controlled cars and in Sportfriends you play a variety of different spots. In these games even though each character has ups and downs (some are faster, some are stronger, some have a better turn radius etc…) after careful testing I have decided that these differences are not impactful enough to base the while balance structure on. In these kinds of nearly fully skill based games regardless of the tools the players have the more experience and skilled one will win against a lesser skilled opponent. Another reason is that the differences between tools are not great enough to offer players different play styles. And this point is one of my design objectives.

Secret Ponchos[15]

Secret Ponchos is a top down arena shooter that allows players to pick from a varied array of different characters to fight with. The maps are constrained and controlled, everything boils down to a one versus one showdown. The game also offers a variety of different characters to act as my different tools. Each character having its own set of play styles and a variety of different tweaking knobs makes this game an ideal candidate for using as the basis of inspiration.

3.1. Proposed Game Design Overview

Based on the design objectives and the preliminary research the optimal way to implement the self-balancing system seems to with an arena based top down shooter. In this game the player will choose from a set of predetermined characters with weaknesses and strengths. The presence of different characters to choose from will give players enough tools to pursue different strategies. The one on one arena based style ensures that the game environment

is controlled and space can be set up so that the matches do not last longer than 3 to 4 minutes increasing iteration and overall feedback. Each character having many attributes makes it much easier to pick which attributes to alter based on the data.

3.2. Technical Details

The game will be developed using Unity3D engine and C#. Unity3D engine gives fine grain control over the development process while increasing the pace by a huge margin compared to not using any game development engine.

The built in networking capabilities of Unity3D helps us make the game not only local multiplayer but also online multiplayer with the hopes of reaching more players and a better, healthier ecosystem.

3.3. Approach and Objectives

The method of examination would be both qualitative and quantitative. After the game has been developed and a base line of balance has been established as deemed fit by the developer experiments can begin. Two version of the game will be deployed, one with automatic balancing and other without.

The objectives of the experiment is to understand a few crucial points quantitatively and qualitatively regarding whether the auto balancing system accomplishes its tasks:

Qualitative Questions

- Does the game feel balanced? Does each of the characters feel as powerful as the other ones? Does the player feel like he can chose any of the options and compete without being hindered?

- Does the game balance change over time in a controlled manner due to the power levels shifting? How does the players react to this change, positive or negative?

- Do the players feel like the addition of automatic balancing has improved the game experience?

Quantitative Questions

- Does the win rates of each character in the auto balanced system have a more even distribution than the non-auto balanced system?

- Does the character pick distribution in the auto balanced system have a more even distribution than the non-auto balanced system?

- Does the auto balancing system create a stable pattern that evolves based on user feedback or will it find a perfect balance and stay at rest?

3.4. Hypotheses

The auto balancing system will result in a better player experience and result in a more even distribution in both character pick rates and individual character win rates while exhibiting a circular pattern in balance changes.

3.5. Challenges

- Small number of players might not result in enough data. In order for the game and the balancing system to work successfully there needs to be a critical number of players. The design will try to minimize the critical number of players.

- Isolating noise from actual usable feedback will be a task that will require some thinking. In addition for that any data acquired will have to be normalized for randomness and player skill. Once again the game design will be tailored in such a way that randomness does not play any role (or plays a miniscule role) in the outcome of each individual game session.

- The game needs to be fun enough that players place the game due to internal motivation not external motivation.

- The feedback has to be normalized based on the experience and skill of the players. With enough players this problem might be solved however in small numbers the skill and experience difference between players have to be taken into account before the feedback is fed into the automation process.

- The design has to allow for both personal skill to play a great part while making sure each character and the attributes they have bring a unique play style to the table.

4. Further Improvements

- Machine learning techniques can be utilized to fine tune the balance if enough data can be extracted from the players.[17]

- Genetic algorithms can be used to create artificial players.[18] This will allow us to change the player perimeters in any shape and form while giving us access to virtually unlimited amounts of data. This can be used to simulate different ecosystems and observing whether the balance stabilizes, becomes a cyclic turn or becomes unpredictable.

- Different instances in different locations throughout the world (Asia, Middle East, United States, Europe) can allow us to look at different user patterns and maybe draw correlation with cultural differences affecting gameplay attitudes.

5. Budget

5.1. Unity3D Professional Edition [16] - 1500$

The success of the research hinges on the game reaching as many players as possible. Acquiring this license would allow me to publish this game on commercial platforms (free of charge) such as steam which would bring an influx of players/testers.

In addition to the licensing benefits the most important tool that comes with the professional edition, and that is not present in the personal edition, is the Unity Analytics Pro tool. This allows the developer to get insights regarding user behavior. This step is would allow me to run more in depth analysis on the players behavior and analyze the success of the automation of balance.

On a technical standpoint Game Performance Reporting and access to Unity Asset Store 11 would increase the quality of the final product.

5.2. Game Cabinet - 750$

As stated before the success of the papers hinges on reaching out to as many players as possible. Creating a standalone game cabinet and placing it anywhere within the NYU Abu Dhabi Saadiyat campus would allow for a steady stream of players and thus user data. This will also allow for a community to be established around the game which will help analyze the effects of experienced players over novice player on the balancing system.

5.3. User Testing - 500$

Even though in an ideal scenario no external motivation is required for the players the play the game however in the earlier steps incentives may be required to get users to test the system. This budget can either be used in lump sum payments to testers or by creating a tournament with a prize pool.

5.4. Outsourcing Art - 500$

As the main developer is not an artist it would be useful to outsource the art design of the game to make the game reach to more players.

References

1. Vicencio-Moreira, Rodrigo, Regan L. Mandryk, and Carl Gutwin. "Now You Can Compete With Anyone." *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. Print.

2. Hunicke, Robin. "The Case for Dynamic Difficulty Adjustment in Games."*Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology - ACE '05*. Print.

3. Hunicke, R. & Chapman, V. (2004). AI for dynamic difficulty adjustment in games.Challenges in game artificial intelligence (AAAI Workshop). Pittsburgh: AAAI Press.

4. "The International 2015." - *Dota 2 Prize Pool Tracker*. N.p., n.d. Web. 24 Nov. 2015.

5. "The Major League of Legend." Riot Games, 2014. Web. 24 Nov. 2015.

6. Booth, Michael. "AI Systems of Left 4 Dead." (n.d.): n. pag. 2009. Web. 24 Nov. 2015.

7. Newheiser, Mark. Playing Fair: A Look at Competition in Gaming. Strange Horizons. March 2009. strangehorizons.com/2009/20090309/newheiser-a.shtml

8. "Bastion FAQ." *Supergiant Games* -. Web. 24 Nov. 2015. <https://www.supergiantgames.com/blog/bastion-faq/>

9. Jaffe, Alexander, Alex Miller, Erik Anderson, Yun-En Liu, Anna Karlin, and Zoran Popovic. "Evaluating Competitive Game Balance with Restricted Play." *Proceedings, The Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Print.

10. Baldwin, Alexander, Daniel Johnson, and Peta A. Wyeth. "The Effect of Multiplayer Dynamic Difficulty Adjustment on the Player Experience of Video Games." *Proceedings of the Extended Abstracts of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI EA '14*. Print.

11. Olesen, Jacob Kaae, Georgios N. Yannakakis, and John Hallam. "Real-time Challenge Balance in an RTS Game Using RtNEAT." *2008 IEEE Symposium on Computational Intelligence and Games*. Print.

12. *Killer Queen Arcade*. Total Entertainment Web. 25 Nov. 2015. <http://killerqueenarcade.com/field-game-rules/>

13. *Rocket League*. Psyonix. Web. 25 Nov. 2015. <http://rocketleague.psyonix.com/>.

14. Wilson, Douglas, and Nils Deneken. *Sportfriends*. Die Gute Fabrik. Web. 25 Nov. 2015. <http://sportsfriendsgame.com/>.

15. *Secret Ponchos*. Swtichblade Monkeys. Web. 25 Nov. 2015. <http://secretponchos.com/>.

16. *Unity3D*. Unity Technologies. Web. 25 Nov. 2015. <https://store.unity3d.com/>.

17. Andrade, Gustavo, Geber Ramalho, and Hugo Santana. "Extending Reinforcement Learning to Provide Dynamic Game Balancing."*Laboratoire D'Informatique De Paris VI*. Print.

18. Leigh, Ryan, Justin Schonfeld, and Sushil J. Louis. "Using Coevolution to Understand and Validate Game Balance in Continuous Games."*Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation - GECCO '08*. Print.