

Analysis and Creation of Skill Rating System in 5v5 First Person Shooter Games

Christopher Boyce

SUPERVISOR: JAMES BANTON

ASSESSOR: BEN WILLIAMS

Contents

Abstract.....	3
Introduction	3
Aims and Objectives.....	3
Literature Review.....	4
Research Methodologies	10
Results and Findings.....	11
Discussion and Analysis.....	13
Conclusion.....	17
Recommendations	18
Appendices.....	18
Appendix A : Python Program using Elo formula to calculate example Elo and Results	18
Appendix B : Results from Microsoft predictive performance Elo VS TrueSkill (Herbrich, et al., 2006)	18
Appendix C : Example Of Bell Curve.....	19
Appendix D : 343 Industries Research into players leaving when Team Skill Gap is High (Menke, 2020)	19
Appendix E : 343 Industries Research into players leaving when Personal Skill Gap is High (Menke, 2020)	20
Appendix F : TrueSkill Preliminary Test Data	20
Scatter Graph	20
Raw Data	21
Normality Test.....	21
Appendix G : TrueSkill Data.....	22
Raw Data	22
Normality Test.....	22
Scatter Graphs.....	23
Appendix H: Bot 99 Inaccurate results of TrueSkill Test 2 for the use in Graphing lower skilled bots.	23
Appendix I : Elo Data.....	24
Raw Data	24
Normality Test.....	24
Scatter Graph	25
Appendix J : Percentile Data	26
Raw Data	26
Normality Test.....	26
Scatter Graphs.....	27

Bibliography 28

Abstract

This research investigates the current Skill Rating system used in 5v5 competitive shooters and analysis and breakdown of Elo, Glicko, Glicko-2, TrueSkill, and TrueSkill-2. Later in the project, an artifact has been produced that is used to collect data on Elo and TrueSkill and a 3rd Skill Rating system using the research to see if the current solution is a viable option in competitive shooters or if a proprietary option for each game is better. The research concludes that the tested algorithms all perform well but there is a clear advantage to performance-based Skill Rating systems that take into account kills and discussion into other performance stats and how this could benefit future Skill Rating Systems.

Introduction

Competitive shooter games have been around for over two decades and have become one of the biggest genres in gaming, during this time titles such as Counter-Strike, Rainbow Six Siege and newer titles such as Valorant have cultured a large player base which and drawn in by their competitive nature. Each game queued utilizes Skill Rating systems to match you against similarly skilled opponents to create a competitive environment for each player to show their skills. This incentivizes players to carry on playing and improve their skill and eventually master the game (Ebtakar & Liu, 2021). The research project is going to explore and discuss current systems floors and benefits with the outcome to create a fair and robust system. These systems need to work due to thousands of players want to play a fair game as well as prove their abilities and climb the ladders of the rating system. The ranking system also has a major impact on retention due to people wanting to become better. At this current point, there are many methods to do this that will be discussed but not a definitive best solution, this research topic is going to see if there is a best or if each version is viable to implement into a game.

Due to the large scope of this topic, this investigation research will be focused on First Person Shooter due to the similar mechanics throughout each game, therefore, making systems easier to compare between games, but this research can be used in other game modes such as Multiplayer Online Battle Arenas. As well as this the project and artifact will be looking at 5v5 games due to the current climate of competitive shooters majority of competitive shooters use small to medium size teams to compete at ranked play. 5v5 shooters come in many forms such as tactical shooters where a team has an objective normally attacking or defending an objective, they must then duel using teamwork strategy and tools supplied to complete the task. Normally this format of game has a set round timer to complete the objective and when it is completed the round resets and the players start again. The game ends once a set score has been reached. Counter-Strike is the most basic example of this with the objective being to plant a bomb from the Terrorist side and the CT trying to defend two bomb sites (Dyer, 2015).

Aims and Objectives

The first aim of the project is going to look into current solutions that have been used in real-world and video game scenarios. It is going to look at the positives and negatives of each one as well as show real-world data when they have been used. The systems when possible are going to be broken down and the math behind the algorithm is going to be analyzed and explained to show how each system works. Once the research into each system is done. Elo and TrueSkill are going to be implemented into an artifact and data from the system are going to be collected and compared to see the benefits of the system using personally collected data. Finally using all the research, a new skill rating system is going to be developed and used to compare against the Elo and TrueSkill

systems to see if a novel solution can perform better or worse. The data collected and analyzed and made into digestible graphs to be able to compare the data easily and form an outcome of this research.

Literature Review

One of the most important concept that is needed when looking into Skill Rating systems is the Bell Curve or Normal Distribution. The curve is a symmetrical curve that represents a bell shape(Appendix C). This curve represents the majority of data points being close together, with fewer data points being extremely high or lower performers. This bell curve has been shown to represent similar skill distributions in competitive video games as well as in other games such as Chess using this as the bases of their rating system. This happens because of the learning curve of some games and over time a smaller and smaller population will gain these skills due to several factors such as natural ability and time that must be spent learning these skills to progress.

One of the most famous rating systems currently used is ELO (Elo, 1986). This system uses the principles of a numerical representation of a player's skill. This numerical system then breaks up into classes of players. These in chess are represented by letters, E being the lowest at around one thousand Elo score to A being the highest, after two thousand Elo score. Pass two thousand the system starts to name classes from "National Candidate Master" to "Grandmaster" (Chess.com, 2020) (Elo, 1986). The system works on finding the probability of the players winning by using the player's existing Elo score and using the formula below to calculate the points gained or lost after the match. The elo system also uses a K value, this value is based on the weighting of the event and game the higher the K score the more points for the players to gain or lose. The K score is between 10 and 32 when used in Chess (Elo, 1986).

$$E_A = 1/(1 + 10^{(R_B - R_A)/400})$$

$$E_B = 1/(1 + 10^{(R_B - R_A)/400})$$

This formula takes the 2 skill ratings (Ra and Rb) which are the current Elo scores of the players. It then plugs them into the formula to work out the percentage chance of them winning. Which is represented by Ea and Eb. These percentages are then either added or taken away depending on a win or loss and multiplied by the K score. The Ra and Rb score is then amended by these amounts to giving the new skill rating. An example of this formula is displayed in Appendix A where Player A with 1500 Elo plays against Player B with 1600 Elo. In the example Player B wins and the formula is used resulting in Player B gaining Elo Score.

The advantages of Elo include that it is easy to interpret a player's skill. Higher the numbers the better the player which is an important part when building a skill rating system (Izquierdo, 2019). The system also is robust and works proven by its longevity and its use of it in international laws and economic laws (Edelkamp, 2021) (Elo, 1986). Elo splitting the player base into classes also allows for fair and more even matching because it will be placed with people with similar skill ratings as each other, meaning matches should be fair and competitive. The simplicity of the system is one of its greatest advantages, because of the easy-to-use algorithm everyone understands how it works therefore making it feel fair and making the player not feel cheated (Izquierdo, 2019).

Elo also has some disadvantages such as the scaling of the K factor. High-level tournaments in chess have predefined K factors decided and calculated by the US Chess Federation (US Chess Federation, 2013) but using the system outside of chess doesn't have a set algorithm or value that the player can use. Therefore, new rules and systems must be implemented to use Elo in a new sport that could be unbalanced. The other issue is the progression time of the player, the mobility of the system is very

slow compared to newer systems (Izquierdo, 2019) which means players take longer to get to their true skill level when starting, this means that there will be a large number of games before the player gets a fair opponent. Finally, Elo is very limited by the games that it can be used with because the calculation can only be used by 1v1, this means any game that has a third player or team-based game cannot use this system (Guo, 2012). Elo also doesn't incorporate any variation or randomness into the system that happens in games with an element of chance or luck (Edelkamp, 2021) meaning the Elo system cannot be used in these circumstances. Decay of the player's scores doesn't exist in Elo and doesn't even have an effect when the user plays again which causes unfair or uneven games, this is where the Glicko System tries to improve upon the Elo System.

Glicko is the improvement and adaptation of the Elo system, Glicko adds a factor of reliability to the outcomes of the game. The reliability of the player's current skill level is based on how recently they have played and the amount they played. For example, someone who has played recently has a higher chance their true skill level is represented by their skill rating score. Where someone who hasn't played in a while could have improved or deteriorated over this time meaning their score may not reflect their true skill level (Glickman, 1998). This is represented in formulas by the notation RD for "Rating Deviation". The higher the RD indicates the player's score is less reliable and more likely for random outcomes. The rating deviation always decreases when a game is played no matter the outcome, this is because more information and a more accurate score is gained after every game even after a loss. When calculating the Rating Deviation, a predefined rating period has to be set. This is the period of time that the player must play a certain number of games before their RD is increased, the Glicko system works optimally when an average of five to ten games are played in this time period (Glickman, 1998). This would have to be worked out by the administrator of the system.

The "Rating Deviation" is set to 350 when first playing the game. This is due to the game not knowing the base skill of a new player meaning it is at its widest possible range. When the player plays for the first time after a new rating period, they must use this formula to work out their new RD.

$$RD = \min(\sqrt{RD_{old}^2 + c^2t}, 350)$$

The T in the formula is how many rating periods the player has missed if the player hasn't missed any T = 1. C is the constant decided by the administrator of the system. C is calculated by working out how long it would take for a player's skill to reset. Glickman assumed this to be around five years (Glickman, 1998) which in his system means 30 rating periods each period being two months to reset the RD to 350. Glickman also assumes an average player to have an RD of 50. In Glickman's case C = 63.2 but this can change depending on the factors listed.

$$q = \frac{\ln 10}{400} = 0.0057565 \quad 350 = \sqrt{50^2 + c^2(30)} \quad g(RD) = \frac{1}{\sqrt{1 + 3q^2(RD^2)/\pi^2}}$$

Q represents Log (10)/400 and is used in the function to work out g(RD). Once g(RD) is calculated we can use these values to calculate E or in some formulas, it is represented as E(s|r,j,RDj). This then can be used to find d^2 which must include every player competing in the current rating period.

$$E(s|r_0, r_i, RD_i) = \frac{1}{1 + 10^{\left(\frac{g(RD_i)(r_0 - r_i)}{-400}\right)}} \quad d^2 = \left(q^2 \sum_{j=1}^m (g(RD_j))^2 E(s|r, r_j, RD_j) (1 - E(s|r, r_j, RD_j)) \right)^{-1}$$

$$r' = r + \frac{q}{1/RD^2 + 1/d^2} \sum_{j=1}^m g(RD_j) (s_j - E(s|r, r_j, RD_j))$$

After all of the calculations for $g(RD)$, Q , C , E and D have been worked out it is possible to calculate the actual rating of the player after the match. S in the formula represents a win as 1 a draw as 0.5 and a loss as a 0 and M represents a match against a player. Multiple matches against the same player are treated as a new match and have no difference from a different player. Now knowing all the variables and inputting them into the formula it will output a rating similar to the Elo system.

There is also a separate calculation that must be done to get the player's new Rating Deviation this takes the input of the player's current RD and the d values and works out the player's post-game RD from these values. This value will always decrease due to the player rating always being more accurate after a game.

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2}\right)^{-1}}$$

The positive of the Glicko system is that computationally it isn't very demanding and because of this, it can be run after every match is played so the player's skill can be updated constantly (Morrison, 2019) (Herbrich, et al., 2007), which is positive when players want instant feedback from the system. This allows the system to save and process very little data and when making a rating system for video games that run on servers is very helpful to save compute time. Because the system is an extension of the Elo system it has the same benefits of being simple to understand with the player being represented now by two values. The system also has been built where new players can get placed quicker (Glickman, 1998) into their real skill rating and has clear rules when adding them. This means that it is easy to add players over time and place them more efficiently resulting in fewer random matches. As well as this there is control for the administrator of the system to change the values depending on how they want the system to work. This allows the system to be adaptable to different games and gives the system some flexibility.

Disadvantages of the Glicko system include no tracking of player performance or how close the game was. This is a disadvantage because the players will lose the same amount of rating even if the game is close or not. If this was tracked the system could better place people and therefore be more accurate (Morrison, 2019). Glicko still has the issue of only working in a 1v1 scenario meaning that in multiplayer games the mean of the team's score will have to be taken making the game therefore less accurate (Morrison, 2019). This also doesn't take in variety from player to player so each player will gain or lose the same amount. There is also the issue of stagnation of the ranks when the player's RD is very low. This creates the issue of the player's improvement being very slow and it would be an advantage for the player to have a higher RD to move up quicker which could be exploited. Glickman has the solution to this issue by not allowing a player's RD to go below thirty (Glickman, 1998).

The Glicko System also gains more improvements over time with the Glicko-2 System bringing more factors into the calculation. This version of the calculation incorporates the player's volatility into it. This is important due to player's performance will never truly be consistent (Glickman, 2022). This is another value when lower shows that the player is more consistent in their performance therefore improvement can be measured more accurately and vice versa. Like the Glicko system, the volatility will increase if the player doesn't compete over a rating period as the system expects change over time. It is still recommended a rating period would cover five to ten games (Rotou, et al., 2015).

TrueSkill is one of the most well-known systems in the Video Games industry being developed by Microsoft in 2005 for the beta of Halo 2 (Herbrich, et al., 2006) and used after on a large amount of Xbox Live games. The reason TrueSkill was created was to deal with rating team-based games with

potentially multiple teams. TrueSkill still uses a Gaussian distribution system by Microsoft which uses the values from Zero to Fifty to rate the player and new players start in the middle at Twenty-Five which in the formulas are represented by μ . The system skill incorporates deviation of the player skill like the Glicko system and is set to 8.333 as a new player (Dehpanah, et al., 2020). This is the formula represented by σ . This allows the system to know if the victory is unlikely and calculate the difference it will make to the player's rating.

TrueSkill is shown to have fewer errors compared to Elo and Glicko when predicting games as shown via sample data pulled from PUBG: BATTLEGROUNDS (KRAFTON. Inc, 2017). This was measured by calculating the "Mean Absolute Error" of over One Hundred Thousand games and over Two Million players (Dehpanah, et al., 2020). As well as this the data shows that TrueSkill is the best at rating new players as well as frequent players and described as the "best performer overall" (Dehpanah, et al., 2020). This data set being large cannot be fully relied upon due to the game being a "Battle Royal" genre meaning it is Free For All and therefore may not translate to other games. This point is once again proven in the data for Halo 2 when measuring the match quality. In data produced by Microsoft, it shows that the system is better than Elo at finding matches that are considered competitive but only in "Free For All" and "Head to Head" modes. Using the TrueSkill system meant having closer matches as well as more draws. When moving to "Small Team" game modes it fails to beat Elo. This is possible due to more players meaning more volatility in their skill levels at the time of the games (Herbrich, et al., 2006).

The first version of TrueSkill has a glaring issue. That it assumes that a player's performance is separate from other players within the team and teamplay isn't taken into consideration. An example of this could be having to go in first in a tactical shooter game where you are more likely to get killed first and have less chance to have an even or fair fight. The system doesn't take into account that the player could have sacrificed for the team. The other issue with TrueSkill was during the testing of the system in Halo 2 it showed higher prediction errors when used in smaller teams (Herbrich, et al., 2006). Show in Appendix B. In the case of a 4v4 full run Elo performed better than TrueSkill but it was also shown in closer matches to be more accurate and with a lower error rate. Even taking this into account the difference is very low but not insignificant.

In 2018 Microsoft published Trueskill-2, this system added to the original and looked at the flaws of TrueSkill and tried to build upon them. They had several criteria that they wanted to meet. These were Support for team games, changeable skills rating, and single number representation which was possible in TrueSkill but they also wanted to add additional requirements. (Minka, et al., 2018) Aligned incentives were important because they want the player to work for the overarching goals instead of being selfish and exploiting the systems to gain rating. They also want minimal training data because it is difficult to test the data with systems before releasing it into a game, as well as low computational costs meaning running the system should be cheap for the company to update. Finally, minimal tuning needs to get to work this the game as it is being implemented.

Trueskill-2 considers more data points, such as kills and deaths in addition to the win and loss. This is a dramatic change because it separates the player's performance from the rest of the teams allowing for a more accurate representation of the player's skill rating. It also handles player quitting or disconnecting as a loss punishing the player for this to avoid people leaving to avoid rating decreasing. This could have been used previously as an exploit and will now allow the other team to gain the rating they have earned. Trueskill-2 also now uses data from other game modes to more accurately predict where they are going to be placed in a new mode, this is because it is assumed that the player's skills will translate through modes if they are similar for example being 2v2 to 5v5 because the player has the base skills they have developed, and they are now using it in a larger

team game. The system now has a bias toward newer players meaning the first few games have a large swing on the player's rating as the system now understands that a large range of skills is possible. This could mean more accuracy quicker in the player's ranked experience. Finally, the system now takes into account the performance increase when the player is queuing with other people and can balance the teams more effectively. This has been proven as a team that plays together regularly develops a "Collective Intelligence" and skills where they can adapt quickly to a range of different scenarios (Kim, et al., 2017).

With all these new features that have been implemented, Trueskill-2 can predict results more accurately as well as give more accurate skill ratings to the player. This is also shown the improvement of the predictive accuracy being 52% for the original but with this update, it is around 68%. Microsoft implemented Trueskill-2 into Gears of War 4 and believe the system was superior and worked well. Microsoft also uses this skill rating on the player to change the difficulty of the AI in the game to challenge the player. This provides feedback to the designers about the average skill of each player.

Trueskill-2 doesn't come without its disadvantages with the main issue being the complexity and difficulty of implementing it into other games. As well as this Microsoft keeps the inner workings and balancing of the model private. Because it isn't fully disclosed the system is impossible to replicate all that is available is the variables it takes into consideration, this is to stop people from finding exploits in the system or simply coping with their research and work. The metrics they collect also are basic with just Kills and Death collected which in modern competitive shooters isn't the only thing that can be tracked. A lot of FPS games have different roles and people doing jobs that may mean they get fewer kills but are impactful Kills meaning they had more of a swing on the outcome of the game. An example of statistics that are important in first kills or entry kills. The team with the first kill normally gives a large swing on the round and can be an important metric to have. This is shown in the 5v4 stats of HLTV which is the leading stat provider of all CSGO professional games which shows that teams have between 70% to 75% of winning the round after winning the first duel therefore giving weight to the first kill (HLTV, 2023). Trueskill-2 also being such a new system that only a few games developed by Microsoft including Gears of War and Halo currently use the system meaning testing is on only a few sets of data meaning this system could have flaws that haven't currently been found. Other games use similar systems which consider other statistics other than Wins and Losses and are currently used but Trueskill-2 is one of the only documented systems.

When developing a matchmaking system, the Team Skill Gaps need to be measured. A Team Skill Gap is the difference in raw skill between the teams. This can be calculated by finding the average team Skill Rating Score and comparing them. If they are too far apart first of all it feels unfair for the team that has a large disadvantage as well as means they are more likely to disconnect causing a total imbalance in the game's integrity. This is shown in Appendix D which is a study that shows the difference between the skill gaps and the likely hood of a player leaving. This example of Halo 5 shows that once the player is over 400 score there is a high chance that one player in the game is going to leave (Menke, 2020). This data shows that 200 is around the maximum before there is a large jump in someone leaving as they find the game unfair. These numbers are specific to the game of Halo 5 but can be used to show proof that it is an important metric to review when in development.

Personal Skill Gaps are the difference between the player's skill verse the average of the enemies' team average. This shows how successful the player is going to be compared to the enemy team. For example, if it is high the player should have an above-average game because they feel like they are better and statistically are better than the opponents. This is also important to make sure there isn't

a high variation due to the fact they will think that the game is too easy or too hard which can cause players to quit or even stop the play session. The graphic of Appendix E is similar to the previous graph and shows a clear curve when the gap gets higher but a more significant incline once above 500 Rating. It is also discussed the player has a high chance after having a game that was too hard that they will stop the play session which is bad for user retention which is the goal of many games companies (Menke, 2020).

Stacked parties have to be considered, this is due because the skill of the party could inflate the skill and responsiveness of the team due to the increased communication and collective intelligence they have gained from playing before (Kim, et al., 2017). This will allow them to tackle in-game problems more efficiently giving them an advantage. Some games disallow large teams to queue together an example of this is Valorant (2020) which the top ranks you can only queue with one other person at the higher ranks. Another solution to this problem is allowing only parties to queue against other parties this will increase the wait time for everyone as the population queuing at the time due to the matchmaker having to take longer to find a party that works. This could also result in games having higher ping. 343 didn't use any of these solutions in Halo 5 and don't have anything to counteract this and let everyone face everyone (Menke, 2020).

Ping will not affect this research topic but it is something that will need to be considered when deployed into a real game, Halo 5 studies show that there isn't a correlation between the players leaving and having higher ping but it is noted that they have got global restrictions meaning that players can't connect to people in different continents meaning the geographical range isn't large enough to cause issues to the player (Menke, 2020). Other server issues such as stability and pack loss will have a more impactful effect on the player meaning a lag compensation is needed to counteract this issue, but it isn't something this research project is going to investigate.

Wait time for a server or to make a game is also shown not to have any correlation with the player quitting the game early this is covered briefly by the Halo 5 study (Menke, 2020), but it is discussed that the more modes and maps that you let the player select the longer the wait times, as well as a discussion on a minimum number players for a game mode, is allowed for larger based games such as battle royals should always be considered.

One external rating tool that has been created is the HLTV Rating System (HLTV, 2017). This rating system was created in 2010 and rates the performance of professional players' matches. This system uses a rating from 0 to 2 and shows the impact the player had on the match more accurately than Kills / Deaths. The system takes a handful of inputs such as Kill Rating, Survival Rating, Impact Rating, and Damage Rating. It uses all these inputs to balance the rating. All these data points allow the algorithm to reflect how the player performed (HLTV, 2022). The introduction website shows the changes from the 1.0 version to the 2.0 version and shows the difference it will and that the players that have more of an impact will gain more score. These inputs that are given to the system can be implemented into a Skill Rating system and will more accurately reflect the player's performance. One of the most important scores is the Impact Rating, this takes into effect the multi-kills, opening kills, and clutches which can show that a player has performed well in the game and is something that isn't currently tracked in systems built into games. The HLTV still has many issues and has been broken down and analyzed and shows that the system still relies heavily on the Kills / Deaths and a lot of the results can have the same conclusion when other variables are removed which shows the system doesn't have the same complexity that is suggested (Sardegna, 2017). This doesn't mean they don't take all the variables into account but proves they don't have a major swing to the final results.

Research Methodologies

The artifact is going to be investigating current rating systems and comparing them between themselves and a new formula using new data points that have been researched. This is going to be done by simulating a competitive shooter and comparing the results to the expected findings such as the natural distribution curve and analyzing the user's performance and contrasting their giving rank to them. This will be done by using the Elo formula to see the most simplistic ranking system and comparing it to the TrueSkill formula a more detailed system. Then a new formula will be included that is going to take inspiration from the several other factors that have been discussed. This formula will be developed and tuned throughout the process of development to replicate or improve over the other systems named. This means there are going to be three forks of the project made with each one with the same development area such as maps bots and the statistics created for each bot.

The artifact will be using simulated player also known as Artificial Intelligence (AI) to evaluate each system and test the performance of each system. That means there will be multiple skill levels using previously discussed statistics of general player performance and applied to the AI in the system to show unique performance differences like in real life where not all players are at the same skill level, this will distribute the players into a large range. The two main statistics that are going to vary on each bot are accuracy and reaction times. These two have been picked due to them being the core mechanic in a basic first-person shooter and an attribute that can be built into the bot aiming script. This also gives the bots at a lower skill level a chance to still compete with higher skill bots due to the accuracy beginning in a range.

The data that will be collected will be scrutinized using several methods the first will be the Kolmogorov-Smirnov test which if the data that is inputted will follow the normal distribution curve will be done due to the research into player performance fitting to the bell curve (Izquierdo, 2019). This test if performed correctly should show with the Elo and TrueSkill formulas follow the bell curve as they have been tested before but will be used to show evidence of correct data in the artifact's development. When developing the new formula this will show that the formula is considered the natural distribution of player skill and will show evidence of a functional system.

Another data-driven way to test for the normality of the results is to create a histogram of the bot's skill rating. All bots data is going to be sorted into bins depending on the spread of the data and then a histogram will be created. This will then be plotted onto a graph to show the curve that has been created. Either the results will show a similar shape to the bell curve therefore backing up the data or the results lack uniformity with the bell curve and prove the data isn't normal. This will help show if the data collected follows the pattern that is expected or that the data collected is somehow wrong and will have to be scrutinized more.

The system is going to need to follow a structure of ranking and the easiest way to do this is following the current system that has been developed. This will mean having stages and levels associated with the current player rating. An example of this is using the system bronze to diamond and using the numbers one to three to show where on the level they are this will give me fifteen unique placements for each player to be placed along this will have to be set later into the development of the artifacts to make each one have a correct pool size and should also show the bell curve for the amount in each which then provide data to show it is working correctly.

Due to the large nature of the project, there are going to be several assumptions that will have to be made. The first is there isn't going to be any networking and location-based pooling of players due to the simulation being performed locally. This will mean all AI are able if at the correct skill level to play each other but for the outcomes of this study this isn't a necessary step. As well as this it is

going to assume each game will finish with no leaving or disconnects, this allows for a reliable pool of data and means every game can be analyzed without external variables allowing the data not to have outliers. Finally, all AI will be in a solo party and not have an advantage queuing with the same AI this is due to the smaller amount of AI in the data set meaning AI will most likely play with AI of similar skills a lot more than in a real-world setting. The bots also will be all playing the same amount of matches at the same time meaning the Trueskill formula will always have the same Mu on each bot. This wouldn't be replicated in the real world as people will have different numbers of games played and therefore would be different.

Data from each match in the running of the simulation will be taken and plotted in an Excel spreadsheet. This will allow for a story of the bots and allow each match to be analyzed and used in the results of this document. This will help to see if there are any inconsistencies in data and allow for more data to be used to see the success or failings of this research. The data is going to focus on bots 0,50 and 99 and plot all matches onto a line graph to show the progression of these specific bots but even 5 matches a scatter graph from all bots will be created. The scatter graph should show a trend line that follows the bot's programmed skill and the results from the skill rating system. For example, bot 0 should in a perfect run have the highest skill rating with it slowly declining when going down the leaderboard.

The final formula is going to try to predict the player's performance in the match and then at the end use this predicted data to see if they performed to the standard that the bot is meant to. It will then use this as a way of disturbing the Elo that the AI gains from each match. For example, if the highest-skilled bot underperforms in the match but they still win it will give less skill rating to that bot. This is going to be dubbed in this document as "Percentile Skill Rating".

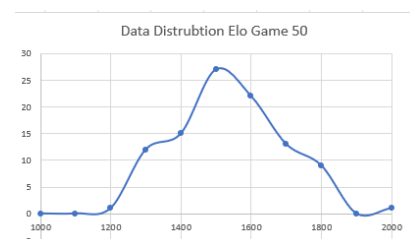
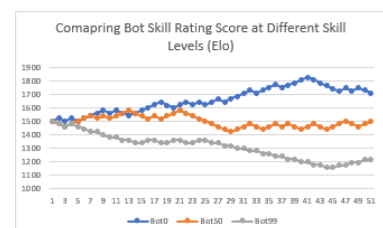
When comparing each skill rating, they are going to compare data to the perfect outcome and see what percentage they get correct. There is also going to be an analysis of each skill rating to see how close they got to this as the results are never going to represent the perfect solution as skill ratings are estimates of the player's skill.

The Unreal Engine is going to be to create this Artifact as many of the features such as Decision trees are built into the engine. As well as this environment can handle a large amount of AI at each level as all 100 bots are going to be performing at the same time. Finally, the use of C++ and blueprints should allow for all functions that need to be created.

Results and Findings

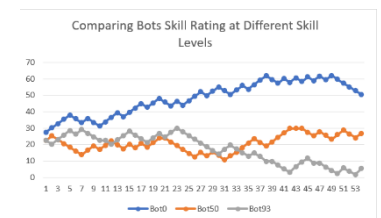
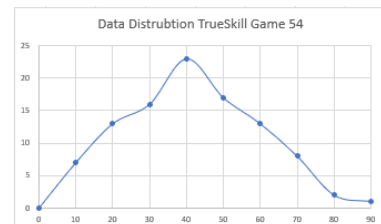
The first test that was run was the Elo system. The K factor of the Elo system was set to 40 to allow an average of 20 Elo scores to be gained or lost over each game.

The system ran for 50 matches to get a large pool of data and to see the long-term effectiveness of the Elo system. All the raw data (Appendix I) were collected and mapped into several different graphs to help interpret the data as well as run through calculations to show the normality of the data. The normality test returned a P-Value of 0.97384 which can be used with the K-S score of 0.04674 to determine the data has some correlation with the normal distribution as a high P-Value shows the percentage chance of the data being normally distributed and the K-S score being low meaning a high chance of the data being normally distributed. The data was also made into a

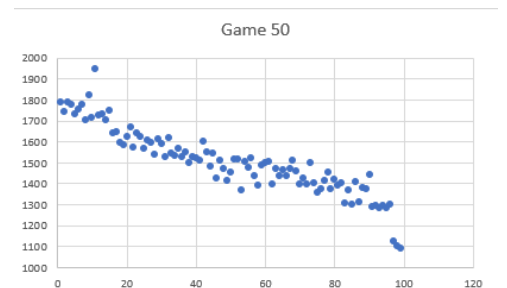


histogram and binned with intervals of 100 to see where the majority of the bot's skill rating was and plotted to a line graph to see. The results of the graph show a similarity to the bell curve, and this is also shown by the 27 of the bots being binned into the 1500 Elo rating, and the further from the midpoint the fewer bots under each bin. Skill rating data was also collected from a high-skilled bot, a medium-skilled bot, and a low-skilled bot to compare their journey throughout the matches and to see the difference, and to see if the system correctly placed each bot. The results show a slow, but significant split of each bot each of them ending with a difference between the skill levels. This data used Bot 0, Bot 50 and Bot 55 which were all very different skill levels to show either end of the spectrum when it comes to players so scatter graphs were collected of all bots every 5 matches and plotted into a scatter graph to see the overall trends of the bots throughout the process. This is shown in Appendix I. The bot's Skill Rating all starts at the 1500 marker but shows after the initial 15 matches which are classed as placement matches its starts to form a trend from the top left to bottom right and this becomes more apparent the more matches played.

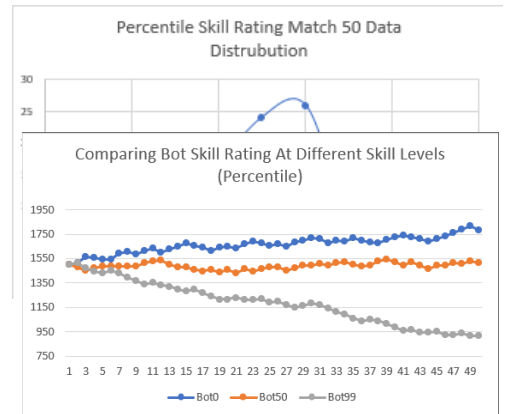
After the initial TrueSkill test of Appendix F, there were some changes. The program was run over 54 games with a higher K factor of 5. This data was to allow for more spread as the initial test results were close together and ran for longer to see the long-term effects of the TrueSkill algorithm and to gather more data points for analysis. The final Mu was collected and run through the same tests as before. This data gave a K-S score of 0.04956 and a P-Value of 0.95639 which mathematically gives a closer alignment with the standard deviation shown in Appendix G. This was also plotted onto a graph using the same methodology of using a histogram. This time the data was plotted from 0 to 90 as the range was much bigger with an interval of 10 between each one. This result was more expected and shows that running the test with a higher K and for a longer period gave a large data pool to analyses. Similarly, before Bot 0, 50, and 93 data was collected and represented as a line graph to allow for an analysis for different skilled bots. Bot 99 was changed due to several factors discussed in Appendix H. Scatter graphs were also created off all bots every 5 matches and at the end to help plot the changes over the matches, this data will allow us to see a correlation of bot placements over the 54 games. Appendix G shows all the raw data that is collected as well as scatter graphs of all the bots Mu every 5 games. This will help us analyze the general trend of the bots and if they are being placed in the correct positions.



The final set of data that was collected was Percentile Skill Rating. This followed the same principles as before. The game was run for 50 matches and all data was recorded shown in Appendix J. The results of the final match were run through a Kolmogorov-Smirnov Test of Normality. This data had the results of a K-S value of 0.07361 and a P-Value of 0.6237 which show the data is far from the normal distribution. This is also shown in the histogram which was made. The data was binned with intervals of 100 between 1000 and 2000 as all results were in this range. The results once plotted clearly show the data doesn't follow the normal distribution but has some characteristics of a bell curve such as the peak but there is a clear lean towards the higher values. Due to this more data was collected from early into the test with data also being gathered from round 30 to see the changes from earlier



in the experiment which is shown in Appendix J. The Bots 0,50 and 90 were also analyzed and plotted onto a line graph to show the journey of different level bots and to see where they ended up. The data objectively shows the increase of the higher skill bots and the reduction of skill rating on the lower skilled bot. It also shows how the average bot stays consistent with the average Elo but does still reduce by a small amount. The data from all bots every 5 matches were also taken and plotted on a scatter graph to show the general trend of all bots and to see if the system is working functionally by placing bots in the correct skill group. This graph is the game from Game 50 and shows the higher skill bots are placed high was a general downward trend showing the bots reducing in skill rating. The progression of these graphs are shown in Appendix J. There are some clear outliers that are significantly higher or lower than expected but this could have been due to several factors including an element of luck with which teams they are placed on or significantly out performing what they are expected to do.



Discussion and Analysis

The Elo data collected has many signs of the system working correctly. The first is the K-S test giving a low score of 0.04674 help prove that the data is close to the normal distribution which is expected for the Elo rating system. This is expected when using this system by the creator Arpad Elo (Elo, 1986). This proves that the results collected are correct and follow the normal distribution pattern. This is also backed up by the Histogram created in Appendix I. The histogram when binned at 100 skill rating intervals shows a similarity between itself and the bell curve. This is shown with the middle Elo 1500 being the peak with 27 of the bots falling under this category. With the next bins showing fall off meaning fewer bots are at 1400 and 1600 making the bell curve. This reflects the theory that people's skill falls onto a bell curve and that the Elo algorithm distributes them accordingly (Izquierdo, 2019). This in no way proves that the bots have been placed in the correct skill group but helps to show the data is correct. One issue that represented itself in the investigation is the sample data isn't very big with the data when binning at 50 intervals showing a much lumpier line as the difference isn't as big. This could be caused by all the bots constantly playing the same similar-level bots causing a cluster of bots all staying around the same numbers. This is also exaggerated due to the bots all playing the same number of games due to the experiment being tested that way and more divergence would happen in the real world as people will encounter a large pool of players. A larger data set theoretically has reduced this lumpiness on the line but more tests would have to be done to prove this.

The analysis of Bot 0, 50, and 99 under Appendix I helped prove that the system is placing bots at the correct skill rating. The data shows Bot 0 (Blue Line) gain skill rating consistently due to it winning the games with a downturn in winning closer to the end most likely by being placed with bots of similar levels and not being able to carry other Bots to the victory as they are all skilled players. Bot 50 (Orange Line) is shown to increase and decrease throughout the process but later becomes more stable. This could be interpreted as the placement matches being sporadic with a large mix of skill groups but slowly over time becoming fairer. This data is consistent with what should be expected from a skill rating system and therefore helps prove the system is working. Finally, Bot 99 shows a consistent drop in skill rating due to it being worse than other bots. From

game 43 the bot does start to consistently win games and this could be the same effect as before with it finally being able to play games with bots at a fair skill level. The data is consistent with what a skill rating system should look like but does prove that the Elo system does take a long to split bots into fair skill levels that they have competitive games as discussed before (Izquierdo, 2019). This mobility of the system means that if the Bots skill did improve significantly, it would take a long time to climb the ladder as the system does look at personal performance. The other notable outcome would be the Skill Rating from top to bottom only being 800 score after 50 matches. This could be due to the number of matches played but also could be to the K factor. This sample used a K factor of 40 which would be considered high in chess but for this experiment could have been higher to get a large range of results. This would have to be investigated to see for differing results.

The scatter graphs created of all bots every 5 games were made to show the correlation between Skill Rating and how good the bot is. The first 10 games clearly show the placement matches scattering the data as many of the games have unpredictable results as the bots are mixed the most. Which will be normal due to the population all starting at the same skill rating at the same time. Around game 25 you can see the data starting to form a trend from the top left down which is to be expected, with this slowly becoming more prominent over the matches. This data shows the system is placing the bots correctly obviously with some outliers which could simply be put down to lucky placement in each team which is bound to happen in a data set of 100 samples. This data concretely proves the system's fundamental part is working as spoken about by Josh Menke at GDC using a similar method (Menke, 2017). This data also proves with the outliers that bots can be carried through games even if they have poor performance or luck, this shows the limitation of using Elo for a Team Vs Team game and means all bots gain and lose the same amount. It also shows the slow pace of the system with it taking 20 to 30 matches to start looking like a fair system with it taking all 50 to be completed. Which is a game that could take a player a long time to play if their retention is low. This backs up the research into the Elo system in the Literature review that stated this.

TrueSkill data also shows that the system performed correctly. When using the Kolmogorov-Smirnov Test of Normality it gave a K-S Score of 0.04956 which is below the 0.05 rating (Ghasemi & Zahediasl, 2012) that is seen as the acceptable rate for the data to show it isn't significantly different from the normal distribution. This is also reflected in the histogram in Appendix G which shows a smooth bell curve when binned with intervals of 10. The histogram shows a near-identical reflection of data from the middle point which shows normal distribution this is shown with a low Skewness. The P-Value is also high with a 0.95639 which shows 95 percent of the results are within one standard deviation making a clear peak in the graph. The one issue with this data set is there isn't a cap to Mu which Microsoft did have in the TrueSkill formula. This was due to the high K factor of 5 being used and therefore bots exceeding the 50 normal maximum. Using a lower K factor and implementing a maximum could have replicated more accurate results but would have meant bots reaching the high end would have bunched up and given the bell curve a flick up on the right side. Overall the proof of normality shows the system is functional and accurately reflects the player's ability.

The analysis of Bot 0,50,93 in Appendix G shows a clear sign that the system places each bot differently and in the correct order, but as discussed in Appendix H that there are some clear outliers. Bot 0 had a steady but clear early increase winning a majority of the placement matches with a more steady but consistent increase over the match period. Bot 50 had a dramatic losing streak at the start and got stuck in the lower end of the skilling rating till around game 30 where the Bot was clearly too low and started to increase dramatically working its way back to the middle of the pack. This shows the clear importance of the first few matches and the chance that the player can get stuck with a bad set of teams in a row and have to work their way back up. This data proves

that TrueSkill longevity works well, and the algorithm works well. This could have been a reduced effect if the bot performance was also considered as it would have in TrueSkill 2 (Minka, et al., 2018). For the reason mentioned in Appendix H, bot 99 was a clear outlier but is proof that singular bots can be boosted through the ranks not by having skill but by pure luck of being on good teams. This is proof there is a floor in the TrueSkill system and shows that queueing in groups may carry a player not deserving of the Skill rating up making the game feel unfair or broken. More games should have been played to see if Bot 99 could have carried on increasing or if the system would bring the bot down to where he should have been. The data K value was set high in this testing due to the shortness of the test and wanting the data to have a larger range to review. In the deployment of this, the K value should be much lower so players do not exceed the 50 cap like in this test.

Finally, the Scatter graphs in Appendix G show a clear correlation even after 15 games with a clear trend of higher-skill bots closer to the top in a short period. The spread of these bots is very large but becomes smaller with each interaction of 5 games. By the end of the test, the data shows a clear trend line and clearly shows the system is working and placing bots in theoretically the correct place. With a few stray middle to low-skill bots being pushed to a higher skill level. This may be due to the carry effect that was shown on bot 99 or that they outperformed what was expected of them. The issue could also be part of the matchmaking system. The system finds the 10 players close to each other and then randomizes the teams. This system may not be the most effective as many lower-skill bots may be placed with a high-skill bot that could carry them. This would have to be something that is looked into and fixed maybe placing them depending on other factors but heavily leaning on the skill rating. Examples of this could be average Kills per game or winning and losing streaks.

Percentile Skill normality data is much more skewed and doesn't represent a bell curve. This is shown by the P-Value being 0.6237 which is for a high value, and the K-S stat is 0.7361. This mathematics shows the data isn't on a normal distribution. This could be caused by several issues within the algorithm that displaces the bots higher or lower than they should be. Looking at the histogram that was made of the percentile system shows the data has a large increase of Bots in the 1500 and 1600 hundred sections with 24 bots and 26 bots respectively. This shows the system keeps a large amount of players close to the middle of the pack. When going to the Skill rating there is a lot of fluctuation in the graph with the line going up and down which doesn't reflect the bell curve. This doesn't prove the system is a failure but shows that the data doesn't fall on the normal curve. Ideally, this test should be run again to prove that this is consistent with the data but this is a new type of Skill rating.

The analysis of bot 0 shows the rise of the skill rating that is very consistent and increased the bot skill rating over one hundred score in the first 10 matches. The bot consistently rises and when losing doesn't impact the skill rating by a high amount due to the performance of the bot being higher than the rest of the team. Bot 50 shows mid-ground where it stays steady in the middle of the pack with no major peaks and loses, this shows the system working for the bot and shows they are being placed well, when looking closer it shows that the bot never peaks above 1540, and never goes below 1440 which show the consistency of the rating system. It would be an interesting test to see how the bot skill rating reacts once the bot stats get better to see if the system can handle a change in skill as a future test. Bot 99 shows a consistent losing streak and it falls does the rating system with it ending at 915. This shows the system has understood the player is worse and is reflecting this well. It would be interesting to see what would happen to the bot if the test carried on to see what rating it would settle at. All of these bots prove the system is working correctly and placing them in the correct order.

The scatter graph shows similar results to the other rating systems, but the clear trend lines seem to form much more quickly. By game 15, there is a clear downward trend, with most bots being very close together and grouped well. This data becomes more apparent as the game continues. At game 30, the bots seem to split to a higher degree, with some bots overperforming and gaining significantly more skill ratings than others. Closer to the end of the games, the bottom skill bots from 95 to 99 seem to fall off significantly more than any other bots. This shows a consistent losing streak and demonstrates that they are much worse than the next 5 bots above them. This is where a larger pool would allow them to be placed with bots of similar skill, but due to the limited pool, they don't have fair matches. The consistent line shows the system working to its full effect, and looking at the data, there is a low amount of outliers, which gives credit to the system's reliance on performance. This system benefits from the basic nature of the game, with kills being the only factor, and therefore the system performs well in this context. However, it may not be as consistent if the game had more features, such as assists, headshots, and team-based motivation, and the system would need to adapt to this. The data also doesn't have a large range, with all the bot's skill ratings being between 900 and 2000, and a sign that the majority of bots are getting stuck in the 1600 to 1700 range. This could be due to the basic matchmaking system or something with the algorithm, but this was also consistent with Elo. If the test were to be rerun, a higher amount of skill rating should be given to see more of a split from higher to lower skill bots and potentially a more complex matchmaking system that takes into account previous games. Overall, the scatter graph shows the success of this test, and the data reflects this well.

When examining all three outcomes of the normal distribution, TrueSkill clearly displays the most normalized data, indicating that the system is placing bots closer to the bell curve. This does not necessarily mean that it is the best system, but rather reflects the accuracy of the system's approximation to the normal distribution of the bots. This is because people's skill in real life also follows a bell curve, and so did the bot data. Elo also followed this pattern but to a lesser extent. This difference is small but could be due to the matchmaking systems and the use of a rating system designed for 1v1 games in a 5v5 game. However, the variation is not significant, and therefore the data is reliable. This is what is expected from these tested algorithms, which have been used for years and have undergone many tests. The normality test of the Percentile skill shows that the data is not normally distributed and accurately reflects the bot's distribution. This would have to be further investigated to determine if it is consistent across multiple runs, but it also suggests that there may be some rank inflation over time, with the general population slowly moving up the ranks, which would need to be monitored.

Percentile skill, when it comes to the speed of a trend line forming in the scatter graphs, is considerably faster than both TrueSkill and Elo. This shows that the system is the quickest to find bot skill ratings and start dividing them into their respective sections. TrueSkill takes around 20 matches to form the trend, but also has a much larger spread than Percentile. Elo is similar to TrueSkill, taking 20 matches to start the trend line, but also has a large spread. This indicates that many of the bots are placed above or below where they should realistically be, which shows some level of inaccuracy. Overall, by game 50, the trend lines for all 3 algorithms have formed similar shapes, but there is a clear high spread in TrueSkill and Elo, which indicates that the algorithms have placed some bots higher than they should be. However, there is never an instance where a bot is at an extreme difference, which would be classified as 25 percent higher than it should be.

The results from the percentile skill need to be scrutinized because the game is so simple. The only raw metric being recorded and inputted is kills per rounds played. While this simplistic approach works well for this game, it cannot be assumed that it would work as effectively in a game with

objectives or a team-based approach. Another design feature that could have impacted the data is the lobby-making system. In this case, all 100 bots were ordered and queued together with the closest bots possible. However, in a real project, lobbies would be sorted into regions, maps, and team queues, meaning that there would be many more variables to consider. This may result in placing bots with a wider skill rating. Hopefully, a completed game would have more players, which would offset this disadvantage, but further testing would be required.

Conclusion

Throughout this project, several different skill rating systems have been investigated. Two currently used systems and one new system were tested, and the results from all three gave positive data, indicating that all of them are viable options. Elo, being the most basic, gave consistent and reliable data and showed that it could place bots in the correct order within a margin of error. The downside of Elo is the time it takes to do this, as it requires over 20 matches to start showing a clear divide between high and lower-skilled bots. Elo has proved to be reliable but has some main issues: it is not suited for team-based games, and it does not consider player performance, which was proven in the data. Overall, this system is good for games that are less complex but lacks performance-based balancing, which could help place players in a more accurate order. This is why Glicko-2 was developed (Glickman, 1998). Elo also deserves merit for how simple the system is and how easy it is to calculate and implement, which could benefit smaller games that don't rely on advanced matchmaking but still want to place players in skill bands.

TrueSkill has consistently shown results but is slightly more balanced as it takes into account a player's current Mu and Sigma values and compares them to the enemy team. However, the algorithm requires 20 matches before it can accurately reflect a player's skill. TrueSkill was a clear first development step for Microsoft in early multiplayer games, but the improvements in TrueSkill-2 would have clear benefits. Research clearly indicates that rating players based on their performance in matches would give the system more data points to analyze and use for matchmaking. Microsoft owns TrueSkill, so most developers not part of Microsoft take inspiration but rarely use it, which stifles the use of this algorithm in many games. Although TrueSkill is much more complex than the Elo system to implement, the results can vary depending on the game it is used in. The newer TrueSkill-2 algorithm would be a future development to test, but due to the system not being publicly available, it is difficult to recreate and test.

The "Percentile skill" algorithm developed for this project shows great signs of working and is the only algorithm tested that uses performance metrics. This helps prove that this approach is the way forward for skill systems. This system gave the closest results to what was expected, as well as the least amount of outliers. Several improvements could be added, as the algorithm is currently restricted to kills. Comparing these results to older skill rating systems shows that the "Percentile skill" system works well for this game, but it should be reassessed and compared to TrueSkill-2 and Glicko-2 to evaluate its effectiveness more fully.

Overall, this project proves that there are many ways to approach skill rating systems, and it is possible to develop and balance a new system that can produce consistent results. The takeaway is that the objective of the game is crucial to balance the system based on the objectives of the game. These systems also need to take into consideration how matchmaking works and how the skill groups are divided. Many large first-person shooters have a large player base, which means they can place more players together to get a fair game, but this also means that the complexity of the system will be higher.

Recommendations

The large nature of the Skill Rating system there is much to be researched. The first would be to use either real player data or using a real testing audience on the project. This will give merit to the results as there could be any faults or failures using AI. This may also produce issues with finding a large enough group of participants so using real-world data from a popular game may be a solution. This would allow for multiple systems to access the same data collected from each match and make separate Skill Rating Scores, this could then be compared and analyzed. This would allow the algorithm to deal with real data and therefore give more accurate results.

The next is the progression of players' skills, this project kept all bots at the same skill stats and didn't show if the system could handle changes in the player skill. A further study into players' skill level changes and seeing which algorithm would deal with the changes the quickest and accurately would help with future developments as in the real world players' skill changes a lot and therefore would be an interest for game developers to know the best solutions.

Finally, more testing of more complex systems such as TrueSkill-2 or Glicko-2 would be interesting to collect and analyze because they are newer and are balanced using player performance which is clearly shown to have a major impact on the system. From this research, there wasn't a large testing of the systems outside of Microsoft's research, and therefore external testing of the Trueskill-2 algorithm would be important to see how it works and is developed. This is a difficulty due to the code not being publically available but there could be a mockup using the same inputs could be made and tested. Using the combination of TrueSkill-2 and other external inputs a better solution could be created.

Appendices

Appendix A: Python Program using Elo formula to calculate example Elo and Results

```
import math
Ra = 1500
Rb = 1600
K = 10
WhichPlayerWins = 2

def Elo(Ra, Rb, K, WhichPlayerWins):
    Ea = 1.0 / (1 + 1.0 * math.pow(10, 1.0 * (Rb - Ra) / 400))
    Eb = 1.0 / (1 + 1.0 * math.pow(10, 1.0 * (Ra - Rb) / 400))

    print("Player 1 Elo Score", Ra)
    print("Player 2 Elo Score", Rb)

    print("Chance Of Player 1 Winning", Ea)
    print("Chance Of Player 2 Winning ", Eb)

    if (WhichPlayerWins == 1) :
        Ra = Ra + K * (1 - Ea)
        Rb = Rb + K * (0 - Eb)
        print("If Player 1 Wins")
        print("Ra Score : ", round(Ra,1), "Rb Score", round(Rb, 1))
    else:
        Ra = Ra + K * (0 - Ea)
        Rb = Rb + K * (1 - Eb)
        print("If Player 2 Wins")
        print("Ra Score : ", round(Ra,1), "Rb Score", round(Rb, 1))

Elo(Ra,Rb,K,WhichPlayerWins)

Player 1 Elo Score 1500
Player 2 Elo Score 1600
Chance Of Player 1 Winning 0.35993500019711494
Chance Of Player 2 Winning 0.6400649998028851
If Player 2 Wins
Ra Score : 1496.4 Rb Score 1603.6
```

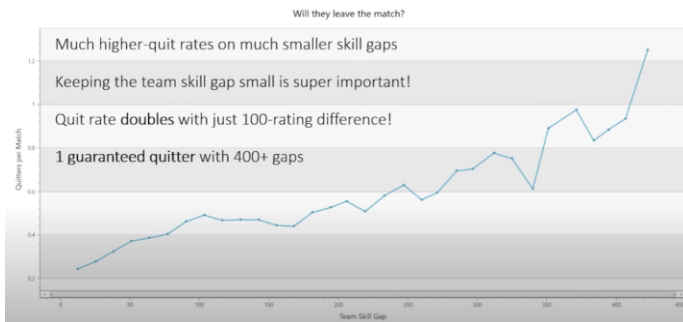
Appendix B: Results from Microsoft predictive performance Elo VS TrueSkill (Herbrich, et al., 2006)

	ELO full	TrueSkill full	ELO "challenged"	TrueSkill "challenged"
Free for All	32.14%	30.82%	38.30%	35.64%
Small Teams	34.92%	35.23%	42.55%	37.17%
Head to Head	33.24%	32.44%	40.57%	30.83%
Large Teams	39.49%	38.15%	44.12%	29.94%

Appendix C: Example Of Bell Curve



Appendix D: 343 Industries Research into players leaving when Team Skill Gap is High (Menke, 2020)



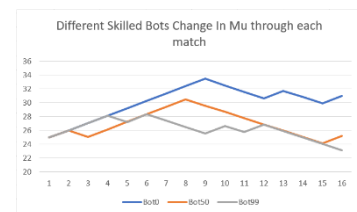
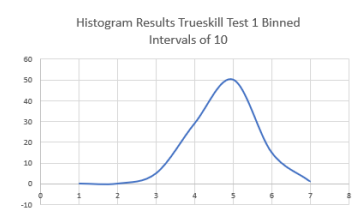
Appendix E: 343 Industries Research into players leaving when Personal Skill Gap is High (Menke, 2020)



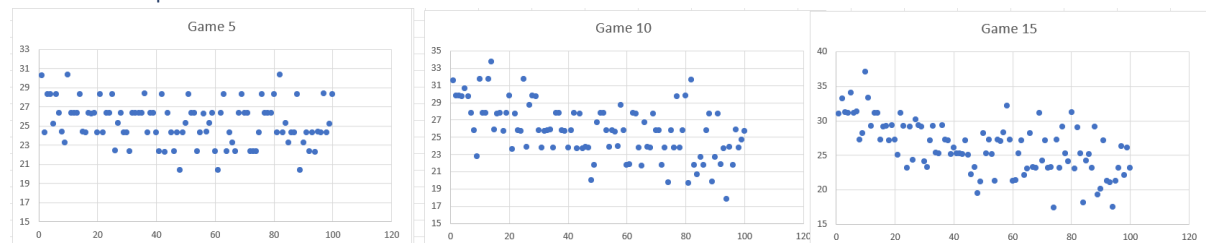
Appendix F: TrueSkill Preliminary Test Data

The first sample of TrueSkill was run over a span of 15 games with a K factor of 2.5 all 100 bots had successful entries and no math errors were found leading to a reliable set of data shown below.

Once the data was collected the last entries of the Mu were run through a Kolmogorov-Smirnov Test of Normality it gave a K-S statistic of 0.07822 and a P-Value 0.54699 as shown in artifact F. The data was also made into a histogram and 2 sets of bins were created. One binned data from 15 to 39 with intervals of 2 and the second binned between 10 and 40 with intervals of 5. The data created a line graph to show the results which then can visually see the curve and compare it to the standard deviation. This graph is the result of the histogram and shows features of the bell curve but doesn't mathematically translate with a low P-Value. Data from the 3 bots were collected and represented on a line graph to show the journey they took through the process. Bot 0 which should represent high-skill players, Bot 50 which represents the average player, and Bot 99 which represents the lower skilled players. Scatter Graph data was also collected to show the bots through the progression of matches. This was taken at intervals of 5 matches and plotted to show all the bots Mu at the current match.



Scatter Graph



Scatter Graphs

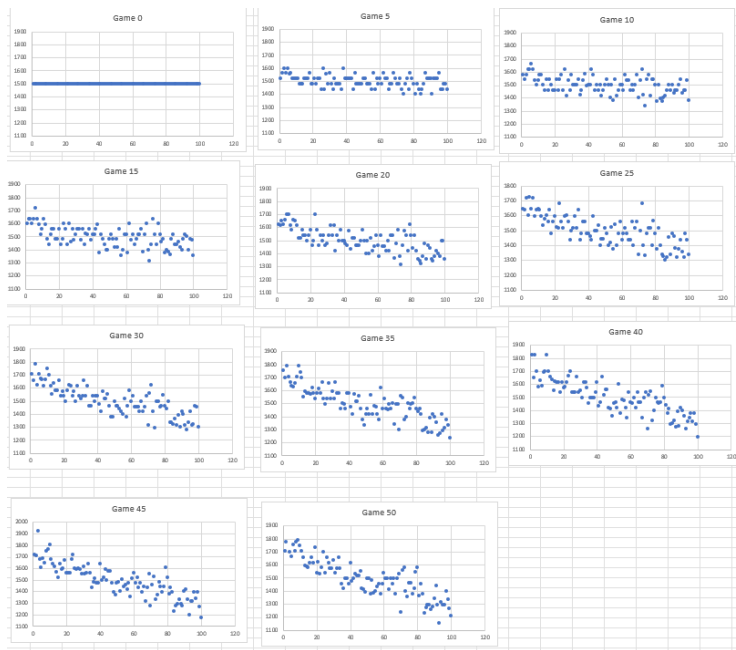


Appendix H: Bot 99 Inaccurate results of TrueSkill Test 2 for the use in Graphing lower skilled bots.

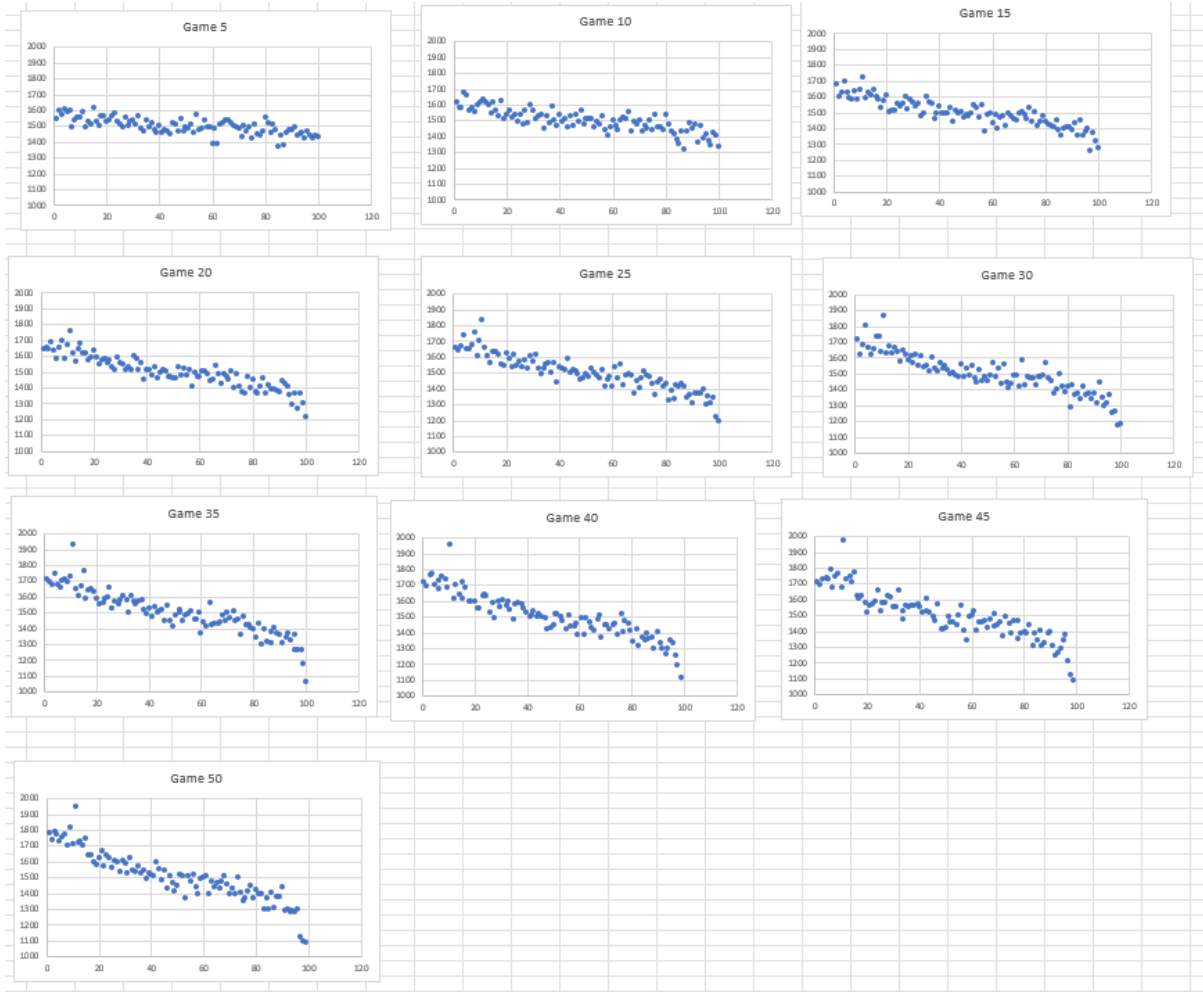
Bot 99 data was collected and was way above average for the bottom 10 bots. This was then broken down using the data collected for each individual bot. It shows the bot winning 26 matches out of 54 which makes the data correct but when studying the Kills and Deaths of the bot it clearly shows a negative KD with it only going positive in 8 matches. This shows the bot was “Carried” through the games and shows a problem with the TrueSkill System. But for the graphs to show a clearer average low-skilled player Bot 93 was selected.

Game	Kills	Deaths	Assists	Score
1	10	15	5	0
2	12	18	3	0
3	8	12	4	0
4	15	20	2	0
5	9	14	6	0
6	11	16	4	0
7	13	19	3	0
8	10	15	5	0
9	14	17	4	0
10	11	14	6	0
11	12	16	4	0
12	10	13	5	0
13	11	15	4	0
14	12	16	3	0
15	13	17	4	0
16	14	18	3	0
17	15	19	4	0
18	16	20	3	0
19	17	21	4	0
20	18	22	3	0
21	19	23	4	0
22	20	24	3	0
23	21	25	4	0
24	22	26	3	0
25	23	27	4	0
26	24	28	3	0
27	25	29	4	0
28	26	30	3	0
29	27	31	4	0
30	28	32	3	0
31	29	33	4	0
32	30	34	3	0
33	31	35	4	0
34	32	36	3	0
35	33	37	4	0
36	34	38	3	0
37	35	39	4	0
38	36	40	3	0
39	37	41	4	0
40	38	42	3	0
41	39	43	4	0
42	40	44	3	0
43	41	45	4	0
44	42	46	3	0
45	43	47	4	0
46	44	48	3	0
47	45	49	4	0
48	46	50	3	0
49	47	51	4	0
50	48	52	3	0
51	49	53	4	0
52	50	54	3	0
53	51	55	4	0
54	52	56	3	0
55	53	57	4	0
56	54	58	3	0
57	55	59	4	0
58	56	60	3	0
59	57	61	4	0
60	58	62	3	0
61	59	63	4	0
62	60	64	3	0
63	61	65	4	0
64	62	66	3	0
65	63	67	4	0
66	64	68	3	0
67	65	69	4	0
68	66	70	3	0
69	67	71	4	0
70	68	72	3	0
71	69	73	4	0
72	70	74	3	0
73	71	75	4	0
74	72	76	3	0
75	73	77	4	0
76	74	78	3	0
77	75	79	4	0
78	76	80	3	0
79	77	81	4	0
80	78	82	3	0
81	79	83	4	0
82	80	84	3	0
83	81	85	4	0
84	82	86	3	0
85	83	87	4	0
86	84	88	3	0
87	85	89	4	0
88	86	90	3	0
89	87	91	4	0
90	88	92	3	0
91	89	93	4	0
92	90	94	3	0
93	91	95	4	0
94	92	96	3	0
95	93	97	4	0
96	94	98	3	0
97	95	99	4	0
98	96	100	3	0
99	97	101	4	0
100	98	102	3	0

Scatter Graph



Scatter Graphs



Bibliography

Chess.com, 2020. *Elo Rating System*. [Online]

Available at: <https://www.chess.com/terms/elo-rating-chess>
[Accessed 28 10 2022].

Dehpanah, A., Ghori, M. F., Gemmell, J. & Mobasher, B., 2020. *The Evaluation of Rating Systems*. [Online]

Available at: <https://arxiv.org/pdf/2008.06787.pdf>
[Accessed 02 12 2022].

Dyer, P., 2015. *Basic CS:GO Tutorial - Beginners Guide*. [Online]

Available at: <https://www.youtube.com/watch?v=a1IK2CKKGzI>
[Accessed 09 05 2023].

Ebtekar, A. & Liu, P., 2021. *Elo-MMR: A Rating System for Massive Multiplayer Competitions*. [Online]

Available at: <https://cs.stanford.edu/people/paulliu/files/www-2021-elor.pdf>
[Accessed 09 05 2023].

Edelkamp, S., 2021. *ELO System for Skat and Other Games of Chance*. [Online]

Available at: <https://arxiv.org/pdf/2104.05422.pdf>
[Accessed 28 10 2022].

Elo, A. E., 1986. The Rating Of Chessplayers Past and Present. In: New York: ARCO PUBLISHING , pp. 3 - 5 .

Ghasemi, A. & Zahediasl, S., 2012. *Normality Tests for Statistical Analysis: A Guide for Non-Statisticians*. [Online]

Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3693611/>
[Accessed 09 05 2023].

Glickman, M. E., 1998. *The Glicko System*. [Online]

Available at: https://hughchristensen.com/papers/academic_papers/glickman1998b.pdf
[Accessed 04 11 2022].

Glickman, M. E., 2022. *Example Of The Glicko-2 System*. [Online]

Available at: <http://glicko.net/glicko/glicko2.pdf>
[Accessed 02 12 2022].

Guo, S., 2012. *Score-Based Bayesian Skill Learning*. [Online]

Available at: https://link.springer.com/content/pdf/10.1007/978-3-642-33460-3_12.pdf
[Accessed 03 11 2022].

Herbrich, R., Danguthier, P. & Minka, T., 2007. *TrueSkill Through Time : Revisiting the History of Chess*. [Online]

Available at: <https://papers.nips.cc/paper/2007/file/9f53d83ec0691550f7d2507d57f4f5a2-Paper.pdf>
[Accessed 08 11 2022].

Herbrich, R., Minka, T. & Graepel, T., 2006. *Trueskill : A Bayesian Skill Rating System*. [Online]

Available at:

<https://proceedings.neurips.cc/paper/2006/file/f44ee263952e65b3610b8ba51229d1f9-Paper.pdf>
[Accessed 02 12 2022].

HLTV, 2017. *INTRODUCING RATING 2.0*. [Online]
Available at: <https://www.hltv.org/news/20695/introducing-rating-20>
[Accessed 09 05 2023].

HLTV, 2022. *How does HLTV rating work? Rating 2.0 explained*. [Online]
Available at: <https://www.youtube.com/watch?v=mymBWc62TcY>
[Accessed 09 05 2023].

HLTV, 2023. *HLTV, Teams, FTU, Bothsides*. [Online]
Available at: <https://www.hltv.org/stats/teams/ftu>
[Accessed 11 01 2023].

Izquierdo, M., 2019. *Ranking Systems: Elo, TrueSkill and Your Own*. [Online]
Available at: <https://www.youtube.com/watch?v=VnOVLBbYIU0&t=1719s>
[Accessed 28 04 2022].

Kim, Y. et al., 2017. *What Makes a Strong Team? Using Collective Intelligence*. [Online]
Available at: <https://dl.acm.org/doi/pdf/10.1145/2998181.2998185>
[Accessed 11 01 2023].

KRAFTON. Inc, 2017. *PUBG: BATTLEGROUNDS*. [Online]
Available at: https://store.steampowered.com/app/578080/PUBG_BATTLEGROUNDS/
[Accessed 02 12 2022].

Menke, J., 2017. *Skill, Matchmaking, and Ranking Systems Design*. [Online]
Available at: <https://www.youtube.com/watch?v=-pglxege-gU>
[Accessed 24 02 2023].

Menke, J., 2020. *Matchmaking for Engagement: Lessons from Halo 5*. [Online]
Available at: <https://www.youtube.com/watch?v=0FoG4Jtpebs>
[Accessed 11 01 2023].

Minka, T., Cleven, R. & Zaykov, Y., 2018. *TrueSkill 2: An improved Bayesian skill rating system*. [Online]
Available at: <https://www.microsoft.com/en-us/research/uploads/prod/2018/03/trueskill2.pdf>
[Accessed 11 01 2023].

Morrison, B., 2019. *Comparing Elo, Glicko, IR Comparing Elo, Glicko, IRT, and Bayesian IR esian IRT Statistical Models T Statistical Models*. [Online]
Available at: <https://scholarworks.uark.edu/cgi/viewcontent.cgi?article=4751&context=etd>
[Accessed 08 11 2022].

Rotou, O., Qian, X. & Davier, M. v., 2015. *Ranking Systems Used in Gaming Assessments and or Competitive Games*. [Online]
Available at: <https://www.ets.org/Media/Research/pdf/RM-15-03.pdf>
[Accessed 02 12 2022].

Sardegna, C., 2017. *Exploring Problems with Counter-Strike Rating Systems*. [Online]
Available at: <https://chrissardegna.com/blog/problems-with-csgo-rating-systems/>
[Accessed 09 05 2023].

US Chess Federation, 2013. *K-Factor Change*. [Online]

Available at: <http://www.uschess.org/index.php/Announcements/K-Factor-Change-May-2013.html>
[Accessed 03 11 2022].