RPG Game Balance Principles and Methodologies: Case Study of *Elden Ring*

Tianyi Xiong

1. Project Statement

The project investigates the logic and balance of role-playing game stat systems through an in-depth case study of Elden Ring, the 2022 Game of the Year. By leveraging in-game data and the game mechanics, analytic methods are applied to examine the game balance, ranging from the player progression curve to the equipment's stat distribution. After that, a battle simulation model was conducted using the Great Lakes Cluster. By incorporating the weapons' stats, damage calculation formula, and the player's behavior, 42,000,000 battles are simulated with the help of the high-performance computing cluster. The simulation serves as an analytical tool to uncover the potential imbalances, ultimately offering insights into how stat systems influence game balance. Throughout the project, I aim to find the principles and methodologies governing an RPG stat system and provide insights for future game design.

2. Background and Related Work

Game balance is a critical aspect of game design, which can be interpreted as "meaningful diversity of gameplay experiences." [1] The core is to create a challenging but rewarding experience by applying appropriate mechanics, difficulty, progression, and fairness.[2] Achieving this balance requires an elaborate adjustment of the game stats and analysis of player interactions. It is affirmed that players desire to be part of the balancing process - investigating and quantifying balance requirements in a player-driven process elevates the very player's role and trailblazes novel approaches of engaging, binding, and tailored games. [3]

Particularly, Role-Playing Games (RPG) rely heavily on stat systems to manage character progression, combat mechanics, and equipment strength. A well-balanced RPG stat system should offer feasible options for different character growth paths, enemies with appropriate levels of challenges, and various alternatives for the equipment. Poor stat balance can lead to dominant strategies that reduce the variety of playstyles among the players.[2]

Elden Ring, the 2022 TGA Game of the Year, serves as an ideal case study due to its comprehensive and widely praised stat system. Various play styles are developed by the players by utilizing different equipment. The game's enemy design and combat mechanics also point out the importance of strategic character growth and gameplay.

Traditional game balance adjustment relied on extensive gameplay testing and the designer's intuition, which lacked accuracy and efficiency. However, quantitative methods have been applied to game balance analysis recently, which involves using statistical models, predictive modeling, and player behavior analysis to fine-tune game stats.[1]

In this project, the Great Lakes High Performing Computing Clusters are leveraged to realize the large-scale simulations, which allow fast and accurate simulation of full HPC applications with minimal modifications to the application source code. [4]

By analyzing Elden Ring's stat system and simulating the battles with the game mechanics, the project reveals the principles and methodologies in the game, offering insights for game designers.

3. Data Sources

(1) Kaggle Dataset: Elden Ring Ultimate Dataset

https://www.kaggle.com/datasets/robikscube/elden-ring-ultimate-dataset

The dataset includes tons of in-game data from Elden Ring, which includes weapons, armor, items, and so on. The dataset is quite reliable and is mainly used to research the logic and methodology of the RPG stat system.

(2) Elden Ring Wiki

https://eldenring.wiki.fextralife.com/Elden+Ring+Wiki

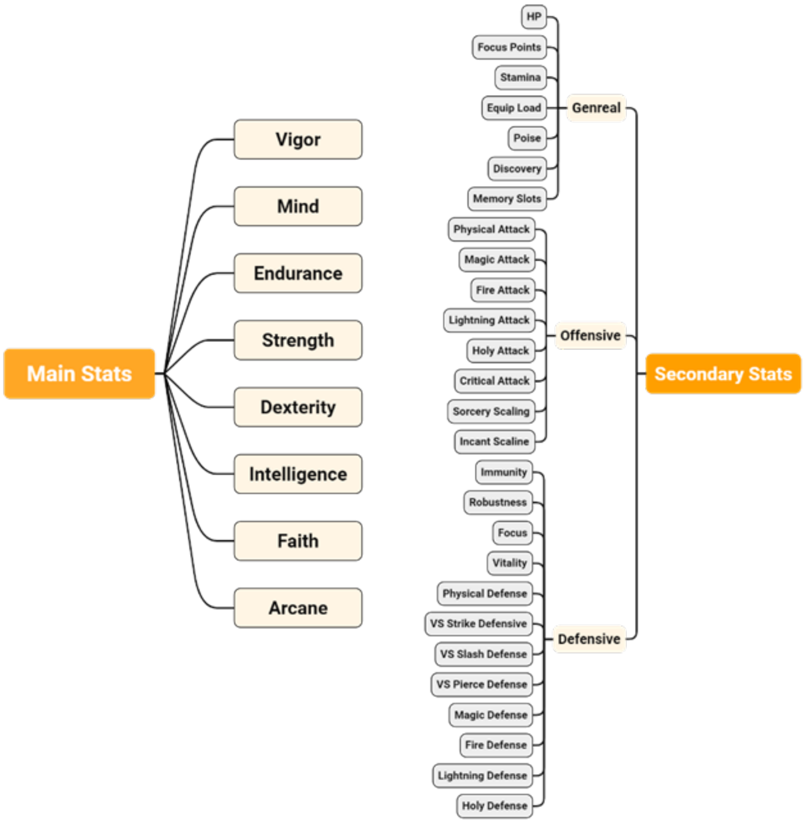
The website serves as a great source to learn the game's mechanics, such as the damage calculation formula and player progression principles.

4. Methodology and Results

(1) Game Stat and Mechanics Analysis

a. Player Character Stats System Overview

In Elden Ring, the player character's stats can be divided into two parts, main stats and secondary stats. The main stats are also called main attributes, which govern all other secondary stats and can be leveled up by spending Runes (an in-game currency). One main stat can impact one or more secondary stats, for example, Vigor not only governs the player's HP, but also affects the fire resistance and immunity stat.



Graph 1: Elden Ring Player Character Stats Overview

b. Balance between character stats: player progression curve

To achieve balance between different main stats and prevent the players from focusing on one specific attribute, the game introduces a concept called **soft-cap.** While the hard caps for all stats are 99, which limits the max value of the stat, the game also diminishes the return from leveling up a specific attribute at a certain point, which is the soft cap. To conclude, the benefit from leveling up an attribute will be less significant than before at a breakpoint. The soft caps of each attribute differ from each other, and each attribute can have more than one soft cap. This principle affects the player characters' progression to a great extent and allows more balanced stats configurations.

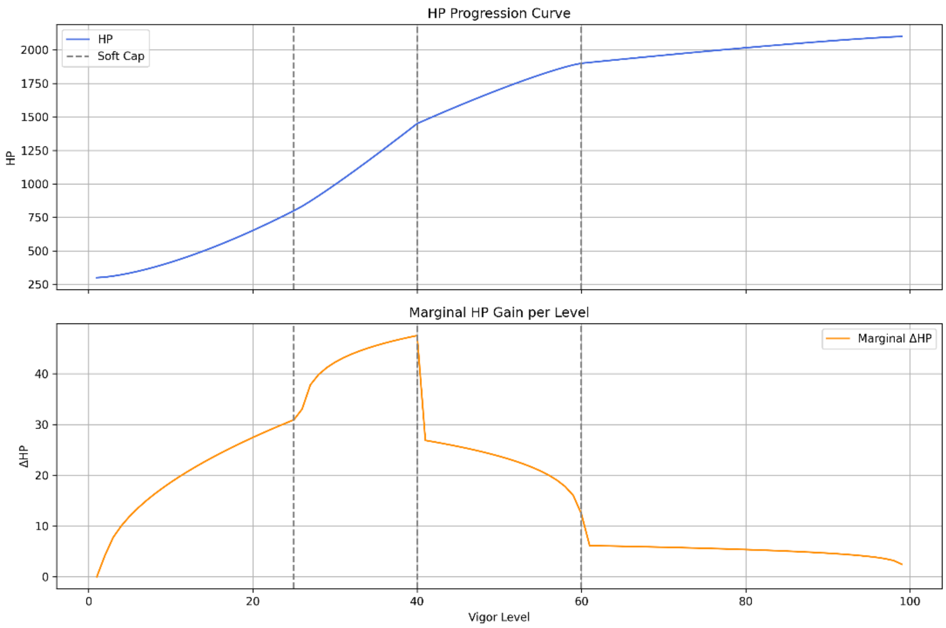
Stat	Soft Cap	Change in gains at soft-cap
Vigor	40/60	up to + 48 / +26 to +13/ +6 to +3 HP
Mind	50/60	up to + 7 / +6 to +4/ +3 to +2 FP
Endurance (Stamina)	15/30/50	+ 1.79 / 1.67 / + 1.25 / 0.31
Endurance (Equp Load)	25/60	+ 1.6 / +1.0 to +1.5 / +1.0 to +1.1

Table 1: Soft Cap Examples

Take "Vigor" for example, the HP gained by leveling up Vigor ranged from 4 to 48 before level 40. But as the players' vigor passes 40, the HP gained will only range from 26 to 13, which will be lower if the players' vigor is more than 60. Since the maximum HP at level 1 is 300, the formula of HP(H) and the Vigor Level (L) can be written as:

$$H = \begin{cases} 300 + 500 \times ((L-1)/24)^{1.5}, & 1 \leq L \leq 25 \\ 800 + 650 \times ((L-25)/15)^{1.1}, & 26 \leq L \leq 40 \\ 1450 + 450 \times (1-(1-(L-40)/20)^{1.2}), & 41 \leq L \leq 60 \\ 1900 + 200 \times (1-(1-(L-60)/39)^{1.2}), & 61 \leq L \leq 99 \end{cases}$$

Formula 1 HP vs Vigor Level



Graph 2: Player Progression Curve & Marginal HP Gain per Level

The graphs provide a visualization of Elden Ring's player progression curve for the Vigor stat and HP. In the top graph, it can be observed that

HP increases rapidly at early Vigor levels, especially up to level 40, but the rate of growth slows noticeably beyond that point. The lower graph, showing marginal HP gain per level, reveals distinct drop-offs in benefit at levels 25, 40, and 60—corresponding to the game's soft cap thresholds. These breakpoints mark where players receive diminishing returns for leveling up Vigor, effectively discouraging overinvestment in a single stat and promoting more balanced character builds. The sharp contrast in marginal gains before and after each soft cap demonstrates how the system subtly guides player behavior while maintaining build freedom, showcasing thoughtful and effective RPG stat design.

c. Game Mechanic Analysis: Damage Calculation

There are eight damage types in the game, including four physical types and four elemental types. The damage calculation follows a uniform formula:

$$Damage = (100\% - Negation) \times Defense\ Multiplier \times Attack\ Power \times Motion\ Value$$

Formula 2: Damage Calculation

i. Motion Values: A hidden motion value associated with each specific attack, usually ranging from 100% to 200%. Generally, the longer an attack animation takes, the higher its motion value.

ii. Attack Power: Attack power for a damage type is based on a number of factors: the weapon's attack stat, the weapon's level and affinity, and the character's levels in any stats that the weapon scales with.

iii. Negation: One of the two defensive stats owned by both the player and enemy. Player's negation for each damage type comes entirely from the equipment. Each armor has its own negations for each damage type. Importantly, damage negations stack multiplicatively, which means the total negation is equal to the product of all damage negation from all equipment you have applied. For example, if the player equips two armors with fire negation 5% and 10% respectively, the total fire negation will be 1- ((1-5%)*(1-10%)) = 14.5%

iv. Defense Multiplier: The defense stat for each damage type is associated with the players' main attributes. For example, physical defense is boosted by strength and magic defense is boosted by Intelligence.

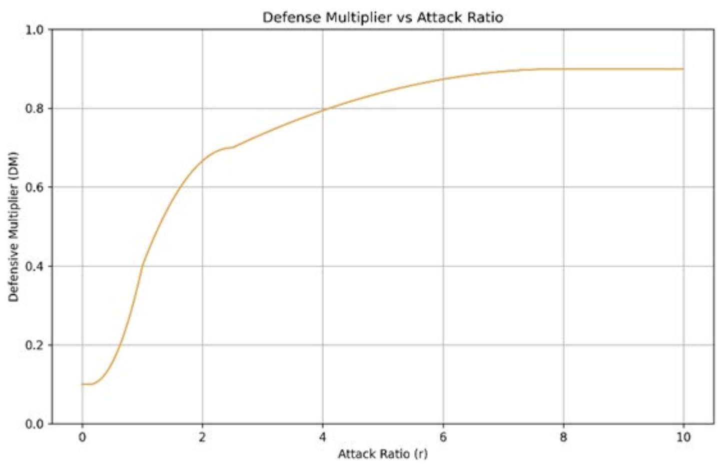
However, the ensure the balance between attack and defense, the game further convert the defense stat to defense multiplier, which is related to another concept named attack ratio (r). The defense multiplier (DM) will change based on the attack ratio. To conclude, this mechanic builds a gradually increasing damage curve as the player's attack becomes stronger, as well as ensuring the damage given if the attack is lower than the opponent's defense.

Attack Ratio (r) = (Attack Power × Motion Value) ÷ Defense

Formula 3: Attack Ratio

$$DM = \begin{cases} 10\%, & r < 0.125 \\ 10\% + (r - 0.125)^2/2.552, & 0.125 \le r \le 1 \\ 70\% - (2.5 - r)^2/7.5, & 1 \le r \le 2.5 \\ 90\% - (8 - r)^2/151.25, & 2.5 \le r \le 8 \\ 90\%, & 8 \le r \end{cases}$$

Formula 4: Defense Multiplier vs Attack Ratio



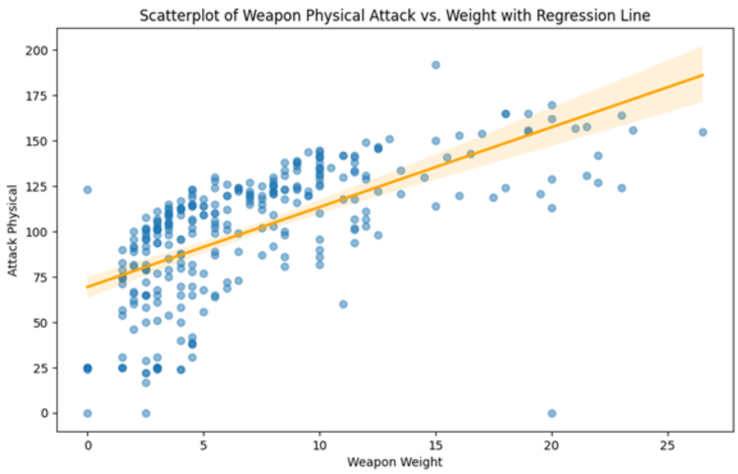
Graph 3: Defense Multiplier vs Attack Ratio

This design ensures the minimum damage given by a character, as well as limiting the maximum damage with a proportional calculation. The nuanced damage progression provides more opportunities for players to combat strong enemies, which will be further examined in the simulation.

d. Balance between equipment: weapons

There are more than 20 weapon categories in Elden Ring, with a total of more than 300 unique weapons. To achieve the balance between the weapons, the game sets the weapon weight and hit per minute stat for each weapon. As the weapon gets heavier, its physical attack power will significantly grow, but the attack speed will decrease as a cost. Besides, since elemental weapons can do both elemental damage, the physical attack power of those weapons is commonly lower than that of pure physical weapons.

To affirm the relationship between the weapons' attributes, regression was applied to the weapon's weight, physical defense, and physical attack power. According to the results, it can be inferred that both physical defense and weapon weight significantly influence physical attack power. Besides, weight has a larger impact on physical attack power (2.75 per unit) than physical defense (0.97 per unit). 53.4% of the variance in physical attack power is explained by these two factors.



Graph 4: Weapon Weight vs Physical Attack

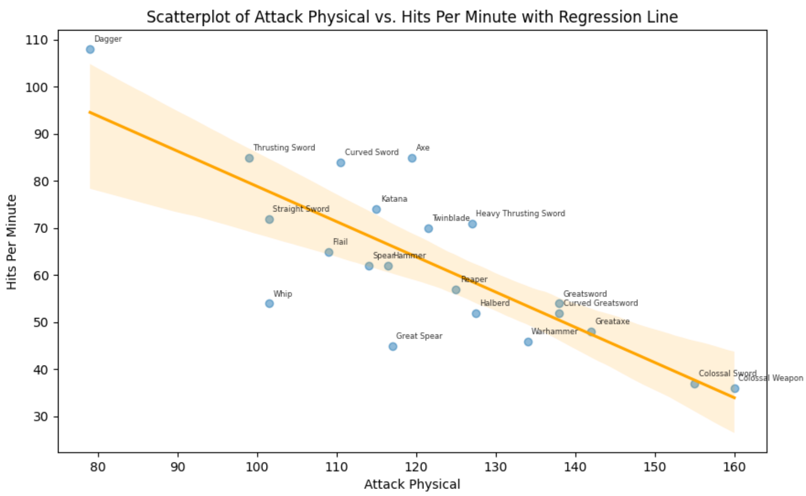
OLS Regression Results			
Dep. Variable:	attack_Phy	R-squared:	0.534
Model:	OLS	Adj. R-squared:	0.531
Method:	Least Squares	F-statistic:	173.6
Date:	Wed, 09 Apr 2025	Prob (F-statistic):	5.69e-51
Time:	15:47:20	Log-Likelihood:	-1414.2
No. Observations:	306	AIC:	2834.
Df Residuals:	303	BIC:	2846.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	42.5802	3.789	11.239	0.000	35.125	50.035
defence_Phy	0.9697	0.106	9.117	0.000	0.760	1.179
weight	2.7474	0.326	8.426	0.000	2.106	3.389

Omnibus:	93.434	Durbin-Watson:	1.159
Prob(Omnibus):	0.000	Jarque-Bera (JB):	521.021
Skew:	-1.127	Prob(JB):	7.27e-114
Kurtosis:	8.982	Cond. No.	117.

Table2: Multiple Regression Result: Attack_Phy ~ Defence_Phy + Weight

In contrast to the positive relationships above, weapons' attack speed seems to have a negative relationship with the attack power. Expressed by hits per minute for each weapon category, it was revealed that the weapons' physical attack decreases as the attack speed increases. This kind of design creates a more balanced weapon system and allows various playstyles among the players.



Graph 5: Hits Per Minute vs. Physical Attack

(2) Battle Simulation: incorporating game stats and player behavior

To further examine the balances of the weapon system and the proportional damage formula, a large-scale simulation was applied with the help of the Great Lakes HPC cluster. A total of 42,000,000 battle scenarios between agents (players) and enemies were simulated using the finite state machine framework, while incorporating both the weapons' stats, damage calculation formula, and the players' behavior. Here are the detailed steps of the simulation.

a. Game Framework: Finite State Machine

A Finite State Machine is a computational model used to represent a system with a limited number of states and well-defined transitions between them. It allowed for realistic and dynamic simulation of turn-based battles. In this simulation, each character operates within the following states:

i. Idle: Waiting to take action

ii. Attack: Executing an attack

iii. Dodge: Attempting to evade an incoming attack

iv. Dead: Defeated and removed from combat

b. Incorporate player behavior and randomization

Each agent (player) was defined by a selected weapon category and specific dodge frequency (sampled from a normal distribution between 0 and 0.8), and standard base stats (HP). Doge frequency is used to simulate different levels of players' abilities to evade the enemies' attacks. Probabilistic dodging behavior is incorporated to reflect the real-time player actions and randomness.

c. Generate the sample enemies

A dataset of 1000 enemies was procedurally generated using normal distributions. To examine the effect of the proportional damage formula, the enemies' defense and attack stats are adjusted to cover all ranges of the proportional damages according to players' median attack and defense. A "defense multiplier" label was assigned to each enemy based on its defense value. This label categorized enemies into five tiers — Very Low, Low, Median, High, and Very High — corresponding to different expected effectiveness of the proportional damage formula (based on the ratio r = player's median attack/enemy's defense):

i. Very Low: $r < 0.125$

ii. Low: $0.125 <= r <= 1$

iii. Median: $1 <= r <=2.5$

iv. High: $2.5 < r <= 8$

v. Very High: $r > 8$

To conclude, for each enemy, five key attributes were sampled:

i. HP: Between 250 and 1000

ii. Attack: Between 20 and 200.

iii. Defense: Between 10 and 1000. The values are

iv. Attack Speed: Between 30 and 120 hits per minute

v. Dodge Frequency: Between 0 and 0.3. This is used to simulate the movement of some "smart" enemies that can dodge the players' attacks.

d. Convert the attack speed

To integrate the attack speed, in other words, the hits per minute stats into the game. Python's heapq module was leveraged, which functions as a priority queue.

```
heapq.heappush(queue, (60 / player.attack_speed, 'player'))
heapq.heappush(queue, (60 / enemy_char.attack_speed, 'enemy'))
```

The next attack time is pushed into the queue. where 60 / attack_speed gives the number of seconds between consecutive attacks. Since heapq automatically retrieves the lowest value first, it ensures that the character with the soonest available action gets to move next. This simulates the impact of different attack speeds for both the player and the enemy.

e. Prepare for the Slurm script

Since there are 21 weapon categories and 20 dodge frequencies, there are 420 unique agents waiting for battle with the generated 1000 enemies. The expected process is that each agent fights each enemy for 100 rounds and records each round's result. To achieve the total of 42,000,000 battles (21 weapons × 20 dodge values × 1000 enemies × 100 rounds), a slurm script is prepared to parallelize and submit the jobs to the Great Lakes HPC cluster. The script ran 10 rounds of battles for each agent and enemy and repeated that 10 times to achieve the 100-round battles.

f. Turn-based combat simulations

The turn-based combat simulations include the following key steps:

i. Both player and enemy are inserted into a heapq with their respective action times.

ii. The queue determines whose turn it is to attack next.

iii. Attacks are attempted: if not dodged, damage is calculated using the nonlinear formula from Elden Ring based on the attack-to-defense ratio.

g. Interpret the outputs

i. Win rate of different players against the enemies with 0 dodge frequency: weapon balances

First, to examine the pure effect of the weapons, all battle results with player dodge frequency 0 are extracted to neglect the effect of players' proficiencies.

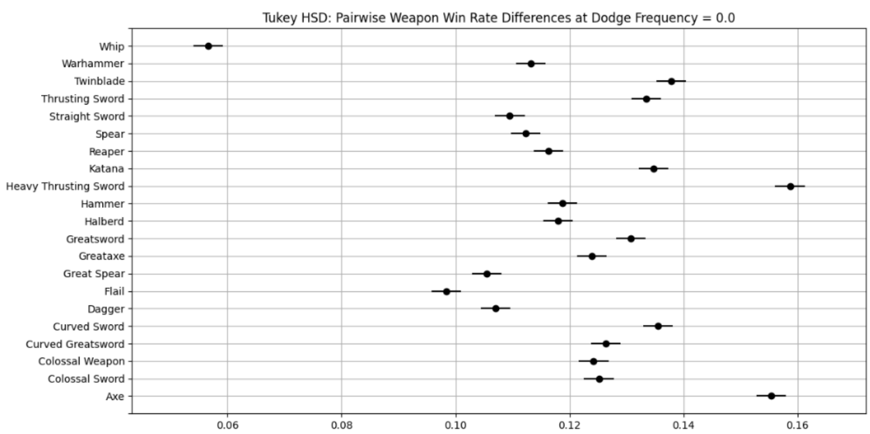
To examine whether there is a statistically significant difference in the mean win rates across different weapons. One-way ANOVA (Analysis of Variance) is applied to the data. And Tukey's HSD is then used to identify which weapons differ and detect the outliers.

Null hypothesis (H_0): All weapons have the same average win rate at dodge = 0.0.

Alternative hypothesis (H_1): At least one weapon has a different average win rate.

As a result, 210 pairs of weapons were examined, and 181 of them significantly differ from each other in terms of win rates. Which means 86.19% of the weapon pairs have significantly different average win rates.

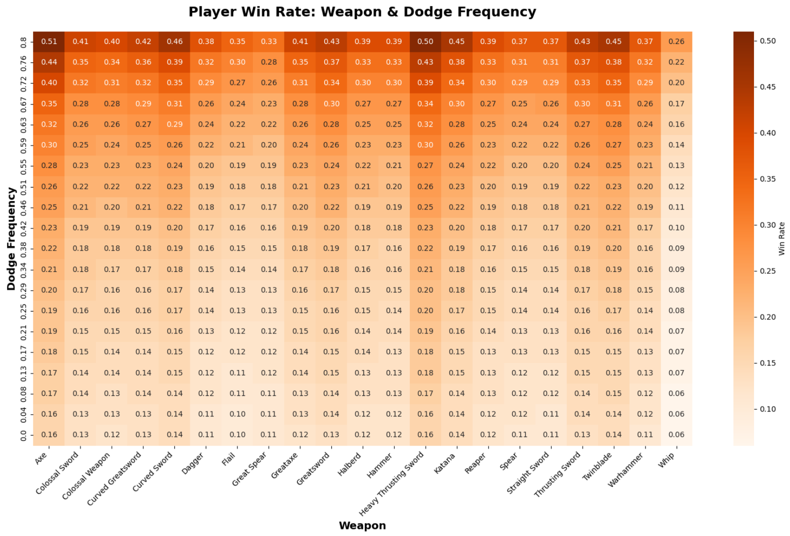
In the Tukey HSD plot, each black dot represents the mean win rate of a weapon, and the horizontal line is the 95% confidence interval for that weapon's average performance across comparisons. According to the graph, it can be interpreted that a heavy thrusting sword has significantly higher win rates than most weapons, making it a potential overperformer. And whip has the lowest average win rate, performing as an underpowered weapon.



Graph 6: Pairwise Weapon Win Rate Differences at Dodge Frequency = 0

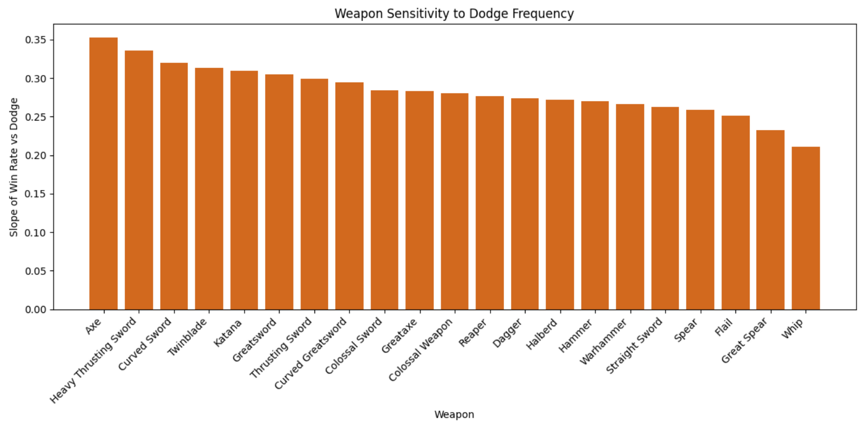
ii. Win rate of different players: weapon + dodge frequency combination

Next, to examine the performance of the players, the weapon and dodge abilities are both integrated to get a heatmap of players' win rates.



Graph 7: Player Win Rate by Weapon & Dodge Frequency

The heatmap illustrates how player win rate varies across different combinations of weapon type and dodge frequencies. A clear upward trend is visible, which means that win rates generally increase with higher dodge frequencies across all weapons, affirming the importance of evasive behavior in such combat games. However, not all weapons benefit equally, which means the sensitivity to dodge varies.

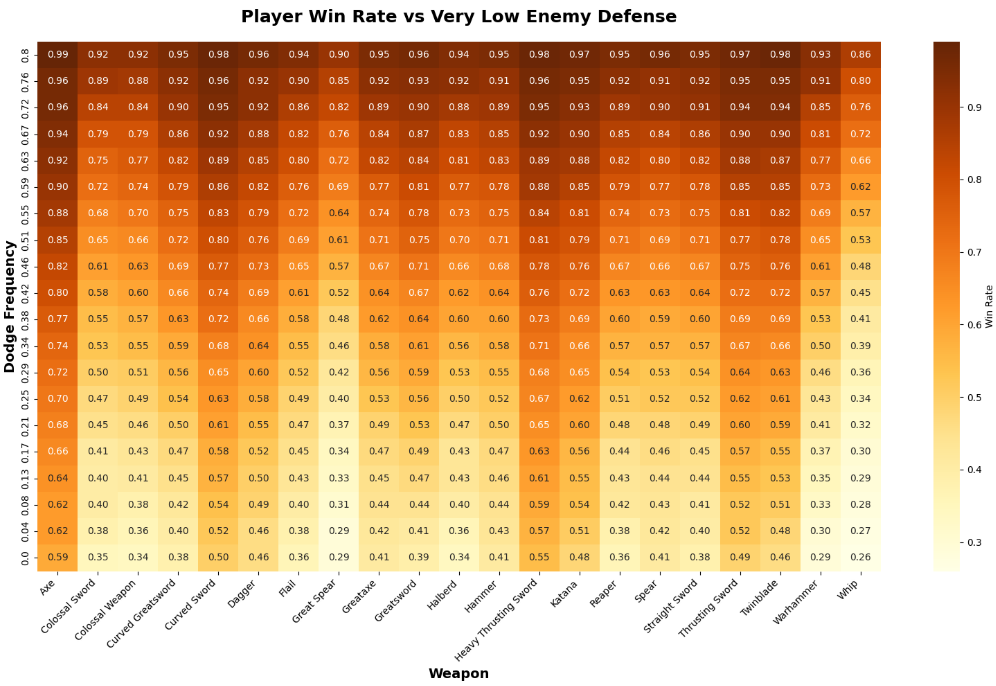


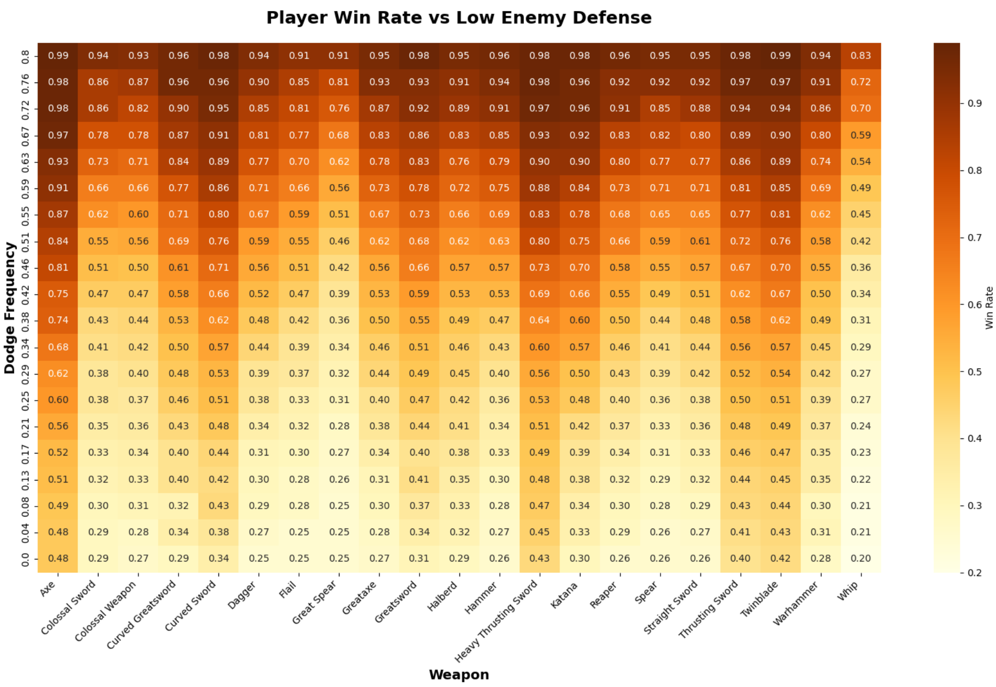
Graph 8: Weapon Sensitivity to Dodge Frequency

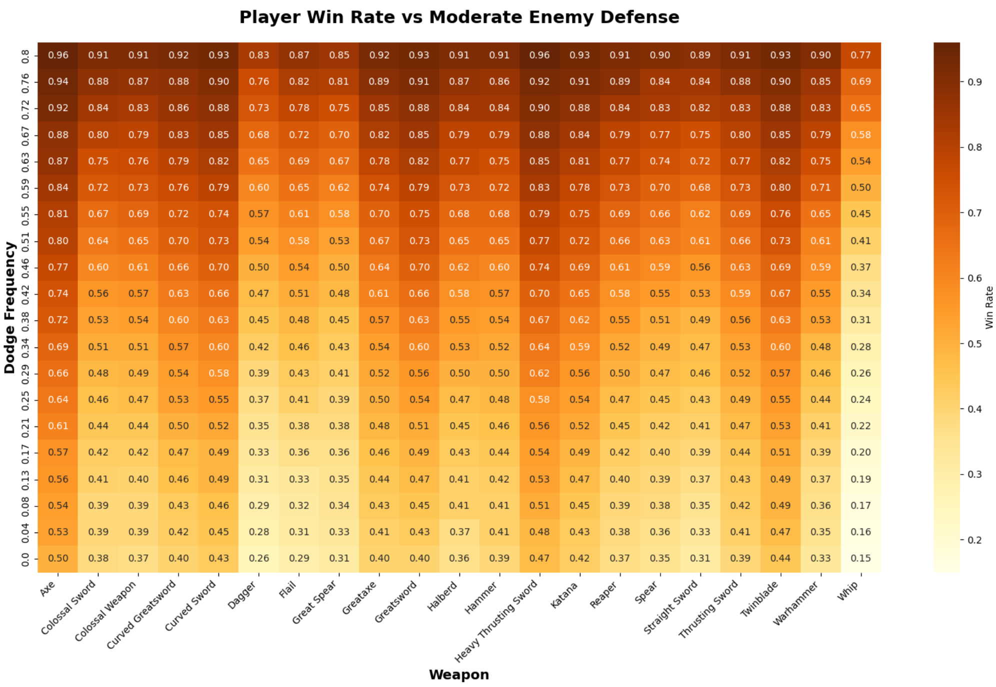
The bar chart quantifies this relationship by measuring the slope of win rate versus dodge frequency for each weapon. Weapons like axes, heavy thrusting swords, and curved swords have the steepest slopes, indicating that they are highly sensitive to dodge. Players using these weapons have a higher increase in win rates with increased dodge ability. On the other end, weapons like the whip and great spear exhibit much flatter slopes, which means the benefit from dodging is limited. It is potentially due to the lower base performance of the weapons themselves, which can hardly be compensated for by players' abilities.

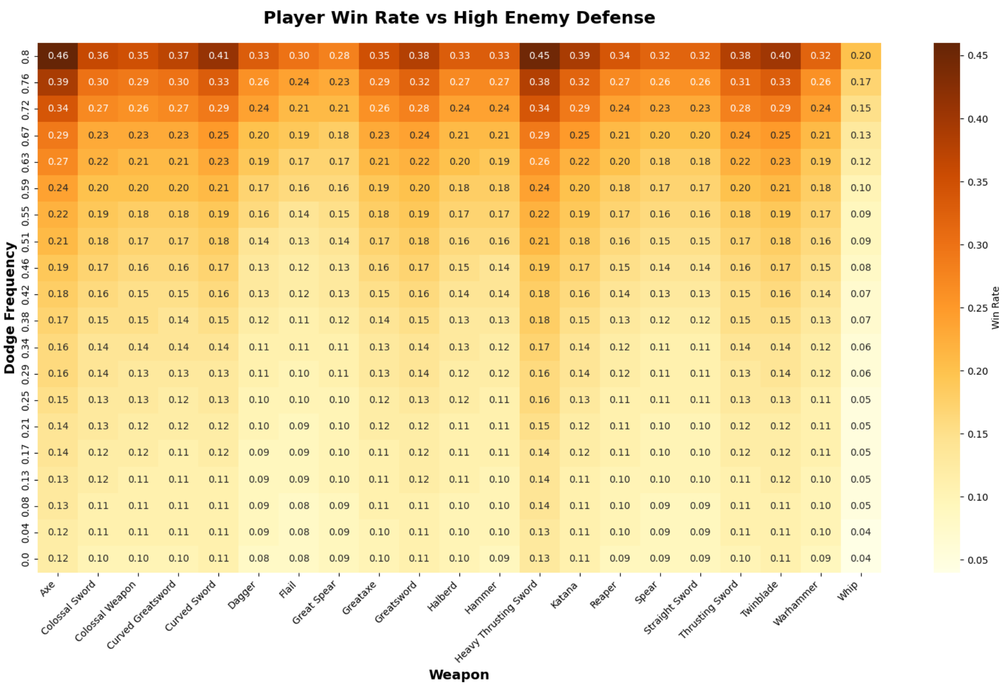
iii. Win rate of the agents against different types of enemies (enemies with different ranges of defenses/attacks)

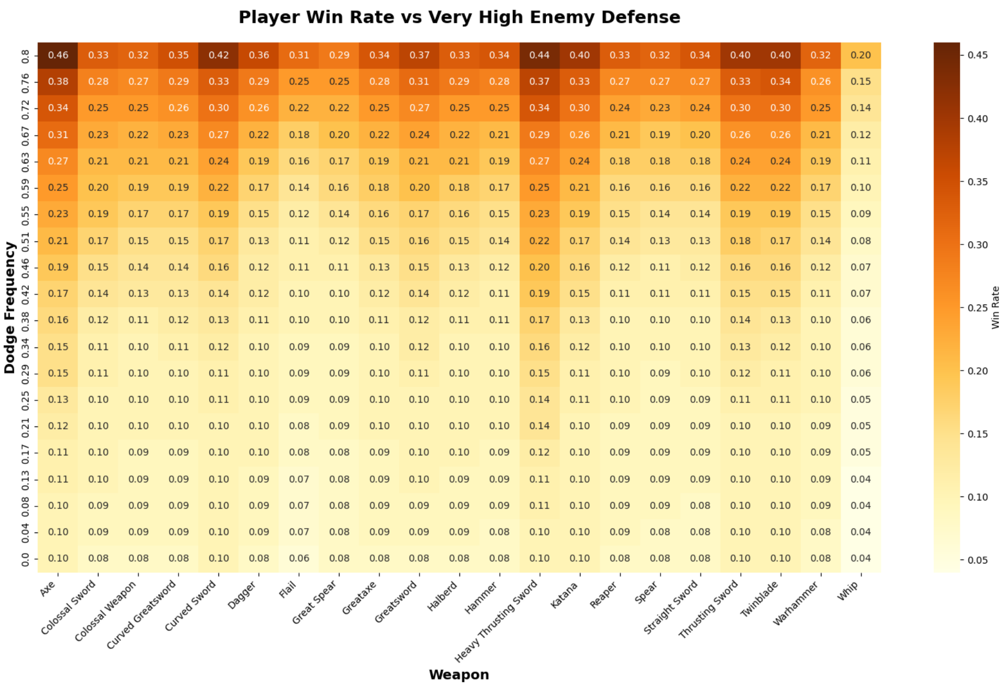
The damage calculation formula serves as an important factor to allow the players to combat different levels of enemies. After separating the enemies by different defense multipliers, from "Very Low" to "Very High", it becomes available to evaluate how different player configurations perform across enemies with different levels of defense.




Player Win Rate vs Low Enemy Defense
Dodge Frequency
Weapon
Win Rate
0.8 0.99 0.94 0.93 0.96 0.98 0.94 0.91 0.91 0.95 0.98 0.95 0.96 0.98 0.98 0.96 0.95 0.95 0.98 0.99 0.94 0.83
0.76 0.98 0.86 0.87 0.96 0.96 0.90 0.85 0.81 0.93 0.93 0.91 0.94 0.98 0.96 0.92 0.92 0.92 0.97 0.97 0.91 0.72
0.72 0.98 0.86 0.82 0.90 0.95 0.85 0.81 0.76 0.87 0.92 0.89 0.91 0.97 0.96 0.91 0.85 0.88 0.94 0.94 0.86 0.70
0.67 0.97 0.78 0.78 0.87 0.91 0.81 0.77 0.68 0.83 0.86 0.83 0.85 0.93 0.92 0.83 0.82 0.80 0.89 0.90 0.80 0.59
0.63 0.93 0.73 0.71 0.84 0.89 0.77 0.70 0.62 0.78 0.83 0.76 0.79 0.90 0.90 0.80 0.77 0.77 0.86 0.89 0.74 0.54
0.59 0.91 0.66 0.66 0.77 0.86 0.71 0.66 0.56 0.73 0.78 0.72 0.75 0.88 0.84 0.73 0.71 0.71 0.81 0.85 0.69 0.49
0.55 0.87 0.62 0.60 0.71 0.80 0.67 0.59 0.51 0.67 0.73 0.66 0.69 0.83 0.78 0.68 0.65 0.65 0.77 0.81 0.62 0.45
0.51 0.84 0.55 0.56 0.69 0.76 0.59 0.55 0.46 0.62 0.68 0.62 0.63 0.80 0.75 0.66 0.59 0.61 0.72 0.76 0.58 0.42
0.46 0.81 0.51 0.50 0.61 0.71 0.56 0.51 0.42 0.56 0.66 0.57 0.57 0.73 0.70 0.58 0.55 0.57 0.67 0.70 0.55 0.36
0.42 0.75 0.47 0.47 0.58 0.66 0.52 0.47 0.39 0.53 0.59 0.53 0.53 0.69 0.66 0.55 0.49 0.51 0.62 0.67 0.50 0.34
0.38 0.74 0.43 0.44 0.53 0.62 0.48 0.42 0.36 0.50 0.55 0.49 0.47 0.64 0.60 0.50 0.44 0.48 0.58 0.62 0.49 0.31
0.34 0.68 0.41 0.42 0.50 0.57 0.44 0.39 0.34 0.46 0.51 0.46 0.43 0.60 0.57 0.46 0.41 0.44 0.56 0.57 0.45 0.29
0.29 0.62 0.38 0.40 0.48 0.53 0.39 0.37 0.32 0.44 0.49 0.45 0.40 0.56 0.50 0.43 0.39 0.42 0.52 0.54 0.42 0.27
0.25 0.60 0.38 0.37 0.46 0.51 0.38 0.33 0.31 0.40 0.47 0.42 0.36 0.53 0.48 0.40 0.36 0.38 0.50 0.51 0.39 0.27
0.21 0.56 0.35 0.36 0.43 0.48 0.34 0.32 0.28 0.38 0.44 0.41 0.34 0.51 0.42 0.37 0.33 0.36 0.48 0.49 0.37 0.24
0.17 0.52 0.33 0.34 0.40 0.44 0.31 0.30 0.27 0.34 0.40 0.38 0.33 0.49 0.39 0.34 0.31 0.33 0.46 0.47 0.35 0.23
0.13 0.51 0.32 0.33 0.40 0.42 0.30 0.28 0.26 0.31 0.41 0.35 0.30 0.48 0.38 0.32 0.29 0.32 0.44 0.45 0.35 0.22
0.08 0.49 0.30 0.31 0.32 0.43 0.29 0.28 0.25 0.30 0.37 0.33 0.28 0.47 0.34 0.30 0.28 0.29 0.43 0.44 0.30 0.21
0.04 0.48 0.29 0.28 0.34 0.38 0.27 0.25 0.25 0.28 0.34 0.32 0.27 0.45 0.33 0.29 0.26 0.27 0.41 0.43 0.31 0.21
0.0 0.48 0.29 0.27 0.29 0.34 0.25 0.25 0.25 0.27 0.31 0.29 0.26 0.43 0.30 0.26 0.26 0.26 0.40 0.42 0.28 0.20
Axe
Colossal Sword
Colossal Weapon
Curved Greatsword
Curved Sword
Dagger
Flail
Great Spear
Greataxe
Greatsword
Halberd
Hammer
Heavy Thrusting Sword
Katana
Reaper
Spear
Straight Sword
Thrusting Sword
Twinblade
Warhammer
Whip
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Player Win Rate vs Moderate Enemy Defense
Dodge Frequency
Weapon
Win Rate
0.8 0.96 0.91 0.91 0.92 0.93 0.83 0.87 0.85 0.92 0.93 0.91 0.91 0.96 0.93 0.91 0.90 0.89 0.91 0.93 0.90 0.77
0.76 0.94 0.88 0.87 0.88 0.90 0.76 0.82 0.81 0.89 0.91 0.87 0.86 0.92 0.91 0.89 0.84 0.84 0.88 0.90 0.85 0.69
0.72 0.92 0.84 0.83 0.86 0.88 0.73 0.78 0.75 0.85 0.88 0.84 0.84 0.90 0.88 0.84 0.83 0.82 0.83 0.88 0.83 0.65
0.67 0.88 0.80 0.79 0.83 0.85 0.68 0.72 0.70 0.82 0.85 0.79 0.79 0.88 0.84 0.79 0.77 0.75 0.80 0.85 0.79 0.58
0.63 0.87 0.75 0.76 0.79 0.82 0.65 0.69 0.67 0.78 0.82 0.77 0.75 0.85 0.81 0.77 0.74 0.72 0.77 0.82 0.75 0.54
0.59 0.84 0.72 0.73 0.76 0.79 0.60 0.65 0.62 0.74 0.79 0.73 0.72 0.83 0.78 0.73 0.70 0.68 0.73 0.80 0.71 0.50
0.55 0.81 0.67 0.69 0.72 0.74 0.57 0.61 0.58 0.70 0.75 0.68 0.68 0.79 0.75 0.69 0.66 0.62 0.69 0.76 0.65 0.45
0.51 0.80 0.64 0.65 0.70 0.73 0.54 0.58 0.53 0.67 0.73 0.65 0.65 0.77 0.72 0.66 0.63 0.61 0.66 0.73 0.61 0.41
0.46 0.77 0.60 0.61 0.66 0.70 0.50 0.54 0.50 0.64 0.70 0.62 0.60 0.74 0.69 0.61 0.59 0.56 0.63 0.69 0.59 0.37
0.42 0.74 0.56 0.57 0.63 0.66 0.47 0.51 0.48 0.61 0.66 0.58 0.57 0.70 0.65 0.58 0.55 0.53 0.59 0.67 0.55 0.34
0.38 0.72 0.53 0.54 0.60 0.63 0.45 0.48 0.45 0.57 0.63 0.55 0.54 0.67 0.62 0.55 0.51 0.49 0.56 0.63 0.53 0.31
0.34 0.69 0.51 0.51 0.57 0.60 0.42 0.46 0.43 0.54 0.60 0.53 0.52 0.64 0.59 0.52 0.49 0.47 0.53 0.60 0.48 0.28
0.29 0.66 0.48 0.49 0.54 0.58 0.39 0.43 0.41 0.52 0.56 0.50 0.50 0.62 0.56 0.50 0.47 0.46 0.52 0.57 0.46 0.26
0.25 0.64 0.46 0.47 0.53 0.55 0.37 0.41 0.39 0.50 0.54 0.47 0.48 0.58 0.54 0.47 0.45 0.43 0.49 0.55 0.44 0.24
0.21 0.61 0.44 0.44 0.50 0.52 0.35 0.38 0.38 0.48 0.51 0.45 0.46 0.56 0.52 0.45 0.42 0.41 0.47 0.53 0.41 0.22
0.17 0.57 0.42 0.42 0.47 0.49 0.33 0.36 0.36 0.46 0.49 0.43 0.44 0.54 0.49 0.42 0.40 0.39 0.44 0.51 0.39 0.20
0.13 0.56 0.41 0.40 0.46 0.49 0.31 0.33 0.35 0.44 0.47 0.41 0.42 0.53 0.47 0.40 0.39 0.37 0.43 0.49 0.37 0.19
0.08 0.54 0.39 0.39 0.43 0.46 0.29 0.32 0.34 0.43 0.45 0.41 0.41 0.51 0.45 0.39 0.38 0.35 0.42 0.49 0.36 0.17
0.04 0.53 0.39 0.39 0.42 0.45 0.28 0.31 0.33 0.41 0.43 0.37 0.41 0.48 0.43 0.38 0.36 0.33 0.41 0.47 0.35 0.16
0.0 0.50 0.38 0.37 0.40 0.43 0.26 0.29 0.31 0.40 0.40 0.36 0.39 0.47 0.42 0.37 0.35 0.31 0.39 0.44 0.33 0.15
Axe
Colossal Sword
Colossal Weapon
Curved Greatsword
Curved Sword
Dagger
Flail
Great Spear
Greataxe
Greatsword
Halberd
Hammer
Heavy Thrusting Sword
Katana
Reaper
Spear
Straight Sword
Thrusting Sword
Twinblade
Warhammer
Whip
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9





Graph 9: Player Win Rates vs Enemies of Different Defense Tiers

Five heatmaps were generated to represent the players' win rates against different enemies of different defense tiers, which reveal a clear and consistent relationship between enemy defense levels, player configurations, and battle outcomes.

Against enemies with very low to low defense, nearly all players have high win rates with high dodge frequencies, often exceeding 70%. This indicates that when enemy defense is weak, even underperforming weapons like the whip or warhammer become viable by improving players’ proficiency. However, as enemy defense increases, the distribution of win rates becomes more stratified. At moderate enemy defense level, strong weapons such as the axe and curved sword maintain superior performance, resulting in win rates over 80% when paired with high dodge frequencies, but weak weapons start to fall behind. When enemies' defense rises to high and very high, win rates drop drastically, with many weapon-dodge combinations falling below 20%, particularly at low dodge frequencies. This pattern reflects the influence of the proportional damage formula, which scales damage based on the ratio between attack and defense (r = attack/defense). When enemy defense is high, the ratio shrinks, pushing damage multipliers toward their lower bounds. In contrast, when the enemy’s defense is low, even modest attack values produce large r values, boosting the multiplier and resulting in high damage. Thus, the formula creates a dynamic environment where players need to consider the relative strength of the enemies instead of their own attack powers. This highlights the formula’s role in emphasizing “risk and return” in game combats, which allows for combating against superior enemies but requires stricter stat configuration and player proficiencies.

5. Findings and Conclusions

(1) Findings

The project employed a combination of quantitative analysis, regression modeling, and large-scale battle simulations to investigate the underlying logic and balance principles in Elden Ring. By analyzing the in-game stat system and game mechanics, several methodologies are revealed that promote the game balance and encourage diversity in game strategy.

The concept of soft caps is used to apply diminishing returns to stat investment, effectively encouraging players to build all-around characters. Besides, the proportional damage formula scales the damage output based on the attack-to-defense ratio, which prevents overpowering stats from dominating. The attributes of the equipment follow the principle of "risk and return", which ensures the viability of the weapons.

The simulation incorporated the weapon stats and player behavior to generate over 42 million combat scenarios. While the game incorporated a proportional damage formula and various weapon designs to encourage diverse playstyles, there are still strong and weak agents that exhibit different levels of win rates. It can be inferred that win rates are not only dependent on raw stats but are influenced by the interaction between equipment stats, game mechanics, and player behavior.

(2) Limitations and Future Works

The simulation model used in the project simplifies the actual gameplay, such as enemy AI variability and status effects, which cannot fully replicate the in-game dynamics. And the enemy's attributes were procedurally generated based on statistical distributions since enemy stats can hardly be derived from the data sources. Future research can address these limitations by incorporating real gameplay and enemy stats to refine the simulation model to align closer to actual combat dynamics.

6. **Statement of Work:** Independent Work

References

[1] Jaffe, A. B. (2013). Understanding Game Balance with Quantitative Methods. http://hdl.handle.net/1773/22797

[2] Schreiber, I., & Brathwaite, B. (2022). Game balance. CRC Press.
(Jaffe, 2013)

[3] Pfau, J., & Seif El-Nasr, M. (2024). On Video Game Balancing: Joining Player- and Data-Driven Analytics. Games: Research and Practice, 2(3), 1–30. https://doi.org/10.1145/3675807

[4] Xu, G., Ibeid, H., Jiang, X., Svilan, V., & Bian, Z. (2020). Simulation-Based Performance Prediction of HPC Applications: A Case Study of HPL. 2020 IEEE/ACM International Workshop on HPC User Support Tools (HUST) and Workshop on Programming and Performance Visualization Tools (ProTools), 81–88. https://doi.org/10.1109/HUSTProtools51951.2020.00016