GAME UNIT BALANCE: A CASE STUDY OF MEDIEVAL II TOTAL WAR

EMRE TOPUZ

AUGUST 2022

GAME UNIT BALANCE: A CASE STUDY OF MEDIEVAL II TOTAL WAR

A THESIS SUBMITTED TO THE

GRADUATE SCHOOL

OF

BAHÇEŞEHİR UNIVERSITY

EMRE TOPUZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF

MASTER OF ARTS

AUGUST 2022

T.C.

BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL

MASTER THESIS APPROVAL FORM

Program Name:	Game Design
Student's Name and Surname:	Emre Topuz
Name Of The Thesis:	Game Unit Balance: A Case Study of Medieval II Total War
Thesis Defense Date:	08.09.2022

This thesis has been approved by the Graduate School which has fulfilled the necessary conditions as Master thesis.

Prof. Dr. Ahmet ÖNCÜ

Institute Director

This thesis was read by us, quality and content as a Master's thesis has been seen and accepted as sufficient.

	Title/Name	**Signature**
Thesis Advisor's	Prof.Dr. Barbaros Bostan	
Member's	Dr. Öğr. Üyesi Ertuğrul Süngü	
Member's	Doç. Dr. Diğdem Sezen	

ETHICAL CONDUCT

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Emre Topuz

Signature:

ABSTRACT

GAME UNIT BALANCE: A CASE STUDY OF MEDIEVAL II TOTAL WAR

Topuz, Emre
Master's Program in Game Design
Supervisor: Prof. Dr. Barbaros Bostan

August 2022, 59 pages

The Total War series are getting more complex in gameplay and strategy since the first game was released in 2000. As each game becomes more ambitious, balance between the game and units continues to get more advanced. The Total War strategy series is interesting because it combines multiple modes of strategic gameplay into a single game. Players manage resources, create a political strategy and fight against hostile factions with large-scale armies. Even though unit attributes of the same category or level are highly similar for all factions, some variables determine the winner of the combats. Combats require controlling individual units and more abstract control of unit types, and managing the combat formations and troop deployments on the battlefield at runtime.

This study aims to examine the balance between units of the factions without implicating external factors like environment, weather, morale, etc., by evaluating and comparing their properties provided by Creative Assembly in the Medieval II Total War game and using the game's battle engine, we will evaluate the effectiveness of the game units in the battlefield and give units rankings according to observed effectiveness and will use the ranking data to understand how unit stats work and establish a Combat prediction tool.

Key Words: Combat Prediction, Game Unit, Balance

ÖZ

GAME UNIT BALANCE: A CASE STUDY OF MEDIEVAL II TOTAL WAR

Topuz, Emre
Game Design Yüksek Lisans Programı
Tez Danışmanı: Prof. Dr. Barbaros Bostan

Ağustos 2022, 59 sayfa

Total War serisi, ilk oyunun 2000 yılında piyasaya sürülmesinden bu yana oyun ve strateji açısından daha karmaşık hale geliyor. Her oyun daha iddialı hale geldikçe, oyun ve birimler arasındaki denge daha da gelişmeye devam ediyor. Total War strateji serisi ilginçtir çünkü birden fazla stratejik oyun modunu tek bir oyunda birleştiriyor. Oyuncular kaynakları yönetir, politik bir strateji oluşturur ve büyük ölçekli ordularla düşman gruplara karşı savaşır. Aynı kategori veya seviyedeki birimlerin özellikleri tüm fraksiyonlar için oldukça benzer olsa da, bazı değişkenler savaşların galibini belirler. Savaşlar, ayrı birimlerin kontrol edilmesini ve birim kategorilerinin daha soyut kontrolünü ve oyun içerisinde savaş alanındaki muharebe oluşumlarını ve birlik konuşlandırmalarını yönetmeyi gerektirir.

Bu çalışma, Medieval II Total War oyununda Creative Assembly'nin sağladığı birim özelliklerini değerlendirip karşılaştırarak ve oyunun savaş motorunu kullanarak fraksiyonların birimleri arasındaki dengeyi çevre, hava, moral vb. dış etkenlerden bağımsız incelemeyi amaçlamaktadır. Savaş alanındaki oyun birimlerinin etkinliğini değerlendireceğiz ve gözlemlenen etkinliğe göre birimlerin sıralamasını vereceğiz ve sıralama verilerini birim istatistiklerinin nasıl çalıştığını anlamak ve bir savaş tahmin aracı oluşturmak için kullanacağız.

Anahtar Kelimeler: Combat Prediction, Game Unit, Balance

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my supervisor Prof. Barbaros BOSTAN for his guidance, advice, criticism, encouragement and insight throughout the research.

I would also like to thank my parents and my second family Reboot Interactive for their great support throughout my life. Without their understanding, and continuous support, I could have never been able to aspire for this level of education and complete this study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

Chapter 1

Introduction

"Game balancing is the fine-tuning phase in which a functioning game is adjusted to be deep, fair and interesting. Balancing is difficult and time-consuming, as designers must repeatedly tweak parameters and run lengthy playtests to evaluate the effects of these changes."(Jaffe, 2013).

Medieval II Total War is a popular strategy game released for Windows on 10 November 2006 ("Medieval II: Total War", 2022). Gameplay is divided between a turn-based strategic campaign and real-time tactical battles. The campaign is set between the years 1080 and 1530. Players assume control of a medieval state, referred to in the game as a faction, and control its government, economy, military, diplomacy, and religion on a map spanning most of Europe, North Africa, and the Middle East. In battles, players control groups of soldiers and engage in combat with enemy forces.

Using the game Medieval II Total War's real-time tactical battles we will evaluate the balance of game units from different factions with various playtests and try to understand whether choosing any faction gives dramatic advantages to players in any era of turn-based strategic campaign.

1.1 Statement of the Problem

Relevant research is mostly focused on tuning the process of game balance with automated systems and predicting combat outcomes by processing various replay datas by AI. "Game tree search algorithms, such as Monte Carlo Tree Search (MCTS), require access to a forward model of the game at hand. However, in some games such forward model is not readily available" (Uriarte & Ontanon, 2017).

Some research presents forward models that learn combat models from replay datas. However those models are hard to use and generally use StarCraft II as a test bed for AI algorithms except requiring replay datas. Therefore those models are even

harder to use in RTS games that have various game unit parameters such as Medieval II Total War.

Game units of Medieval II Total War have more than 16 parameters that relate to effectiveness of the unit in the gameplay. Therefore it's time consuming to analyze each unit stat parameters and understand how they affect the combat in the gameplay. When this is the case, it's also time consuming to do unit comparison to analyze the efficiency of game units.

1.2 Purpose of the Study

We present a less complex traditional method to evaluate balance of game units in the context of Medieval II Total War using the game's battle engine itself.

Using the game's battle engine, we will evaluate the effectiveness of the game units in the battlefield and give units rankings according to observed effectiveness and will use the ranking data to understand how unit stats work and establish a combat prediction tool.

1.3 Research Objectives

- To analyze efficiency between units of different factions
- To understand whether game unit balance is fair for all factions
- To analyze power property differential between unit categories
- To establish combat prediction tool to predict combat outcomes
- To analyze whether faction strength and weaknesses are fair in the context of game units.

Chapter 2

Literature Review

Balance is one of the most important concepts in game design and also the most challenging one for game designers. Any game no matter in which platform they can be played must be fair and all the factors that provide the competition in the game must be carefully inspected and balanced to achieve a fair and winnable game experience. These factors can be matching two players appropriately, levels, weapon and unit properties and many others. When balancing a game, designers and developers must build the game by taking all those factors into consideration and because there are too many variables to consider for achieving a game balanced and fairly competitive, sometimes that process would become too much to overcome. Otherwise the game attracts so many angry and not entertained players.

There are several methods in use at the early and mid development phases to balance game mechanics of a game candidate or an existing game. Even though the outcome of a well-balanced game is “fun”, the parameters that make a game well-balanced are quantitative. Therefore, academic studies about game balancing have mostly focused on numbers to achieve fair and playable gameplays.

2.1 Understanding Game Balance with Quantitative Methods

“I define quantitative balance analysis as the use of quantitative methods in formulating, measuring, and understanding game balance”(Jaffe, 2013, p. 2). Balancing a game is a very time consuming process of game design. Designers try to reach a gameplay that feels good by adjusting game parameters. Evaluating the balance of existing games quantitatively is quite important to make new well balanced games at this point. As computers become more powerful, game developers could be able to record the gameplay of their games at the balance tuning phase or an existing game and process the replay datas faster. By recording gameplays, game developers can observe things like players acting on certain conditions, most chosen paths, weapons or characters, etc.

2.1.1 Problems of observations. Understanding of an existing game's balance is crucial to make new variants of that game or make a new game. Most common approach is to evaluate game balance by observation. However observation alone is not enough to make meaningful inferences about the game or gameplay. The alternative to observation is "Restricted Play" (Jaffe, 2013). Impact is a very important concept when it comes to balancing a game. However, impacts might not present healthy inferences only by itself. Therefore, it should be evaluated very carefully.

2.1.2 Restricted play. "Despite the centrality of fairness as a form of balance, it is by no means the only form, as is made clear by the broad variety of forms described by Jesse Schell, (and paraphrased in chapter 2) from 'skills vs. randomness' to 'short vs. long'. Is there any connection between these diverse forms? I claim that there is. In fact, He claims that many varieties of balance can be reduced to that central form, fairness. The form of reduction is simple: to understand the balance of some dynamic, we may frame it as the fairness of a match between two players. one of whom is restricted in a way that highlights that dynamic. In other words, we hallucinate a player (realistic or otherwise) whose behavior captures that dynamic or the lack thereof. This is what we call restricted play." (Jaffe, 2013, p. 42)

2.2 Key Concepts of Game Balancing

Most common key concept in the area is evaluating the balance system of specific games with automated tools. While evaluating game balance of a subject different methods have been used and observing balance systems, researchers are mostly focused on observing the whole balance system of the subject but not specific factors of those systems like units in those games.

Fairness is another key to a balanced game and it's seen as an utmost priority and every node of the game such as races or factions should be fair no matter at the beginning of the game or at asymmetric game conditions and both players and designers are more satisfied when asymmetric conditions are fair (Jaffe, 2013).

To measure the impact of something in the game, researchers measure the win rate of the player who is restricted in their access to that thing when facing another player without that restriction.

2.3 Evaluating Game Balance Without Players

Knowledge of game balance comes from the gameplay by the designers itself first. Gameplays efficiency is dependent on the observations of it. However, designers are not efficient in imagining various play styles that real players can perform. For this reason, game developers use large groups of playtesters to observe their playstyles and reactions.

In the balancing phase of the game, most common road maps for designers are to do rapid playtestings for discovering major balance issues of the game and having a good gameplay by iterating game parameters but this approach is time consuming. On the other hand, rapid playtestings are limited to designers' own play styles.

“Earlier phase of balancing is not often examined in the literature but is of the utmost importance” (Jaffe, 2013)**.** In the earlier phase of balancing, the road map is to play with game parameters to achieve a good level of gameplay and feeling of the game. However, playing with game parameters requires a huge amount of time for the designers. According to A. Jaffe (2013), at this point “Restricted Play” comes on the stage by providing rigorous, measurable definitions of certain forms of balance.

2.4 Automated Balance Systems

There are automated systems for the use of balancing game parameters such as Monte Carlo Tree Search.

On the other hand automated systems require a simulation environment and replay datas of human play. Also, as mentioned before “Game tree search algorithms, such as Monte Carlo Tree Search (MCTS), require access to a forward model of the game at hand. However, in some games such forward model is not readily available”

(Uriarte & Ontanon, 2017; p. 1**).** Jaffe(2013) built a prototype balance evaluation tool to serve as an application of restricted play. The tool measures behavior and play styles of agents with simultaneous moves.

Some study is determined to allow designers to evaluate the balance of existing games. However, case studies for this ambition is weak because of the following issues:

- The case study games are simpler than the majority of games
- Restriction might lead optimal player to play better or worse compared to real players
- Optimal play may decrease the possibility of win (Jaffe, 2013, p. 54)

2.5 Relevant Studies

Previous research on the topic is generally about how game balance affects the challenge and competition in games. Additionally good depth is important as much as good balance to keep players engaged with the game for years (Sirlin, 2009). There are some studies aimed at reducing the difficulties of game balance in different ways. Most famous approach is evaluating and solving balance problems with automated tools. Additionally in a study about balance balance, patterns of balance system of games observed to learn how game balance is affected by the level, represented as an image, and each team's weapon parameters by using computational model uses convolutional neural networks (Kravalos, Liapis & Yannakakis, 2017).

Main perspective in the area is whether balance systems can be automated and those studies mostly have been used on MMORPG games. In those games each race has specific skills and attributes. In the study Solving the balance problem of massively multiplayer online role-playing games using coevolutionary programming(Chen, Mori & Matsuba, 2014) balancing problems of MMORPG games tried to be solved with coevolutionary design method by including integration with the modified probabilistic incremental program evolution (PIPE) and the cooperative coevolutionary algorithm (CCEA), to solve the balance problem of MMORPGs.

Makin and Bangay(2017) tested whether game units are orthogonally differentiated. Units are considered vectors within an attribute space, thus calculating the rank shows whether orthogonal differentiation is consistent with the game. Also in a relevant study it is researched on the possibilities of expanding an application to the other genres by understanding the established balancing model developed from the case study of fighting games (Han, Lee & Park, 2008) . This study is also highly relevant to this thesis because despite Medieval II Total war being a strategy game, it has similarities with fighting games.

Most relevant work to this study Stanescu, Barriga and Buro(2017) produced a combat predictor formula by choosing unit strength values with two methods. First, they could produce and use a simple formula based on one or more relevant attributes such as unit level, cost, health, and so on. Second, they could directly pick a value for each unit type based mainly on our intuition and understanding of the game. Both methods require long playtest sessions for tuning strength values. This study offered learning unit strength values with automated systems by analyzing human replay datas to reduce the effort of tuning strength values.

2.6 Combat Prediction in RTS Games

Combat prediction is very important for RTS games to achieve fairness between game units by adjusting unit attribute values. Most combat prediction models use AI systems and require simulations to predict combat outcomes. “Prediction using simulations is a popular method, but generally slow and requires extensive coding to model the game engine accurately” **(**Stanescu, Barriga and Buro, 2015, p. 1)

Stanescu, Barriga and Buro(2015) introduces a model based on Lanchester’s Attrition Laws to make combat prediction by learning unit strength values using maximum likelihood estimation from past recorded battles of StarCraft II.

Churcill and Burro(2012) estimate the combat outcome of two armies for node evaluation in their alpha-beta search which selects combat orders for their own troops. Similarly, (Stanescu, Barriga and Buro, 2014) require estimates of combat outcomes

for state evaluation in their hierarchical search framework and use a simulator for this purpose **(**Stanescu, Barriga and Buro, 2015).

Some authors proposed a probabilistic graphical model that can predict the winner of the battle after training on simulated battles (Stanescu et al. 2013). Even if it's a promising model, it works accurately for melee fights but underestimates being able to focus fire in ranged fights **(**Stanescu, Barriga and Buro, 2015).

Another study on this field is Predicting Army Combat Outcomes in StarCraft. This study presents a Bayesian model to predict combat outcomes of game StarCraft battle. Their model predicts the winner of a battle and even damage of opponents against each other by assuming two players fight against each other until one of them loses all their soldiers and using simulations. The model they propose is inspired by the ELO rating system used in Chess (Elo, 1978).

2.7 Gaps in the Literature About Unit Balance

There are many studies focused on simulated systems, replay datas, early phases of development and using StarCraft II as a testbed. Studies for understanding combat systems and unit balance in such games have more complex combat systems such as Total War is lacking.

In this thesis, we will follow some procedures to understand the behavior of the complex game units by observing those in the gameplay and establish a predictor to predict combat outcomes that may help designers to balance game units in their game or game candidates with similar complexity of Medieval II Total War.

Chapter 3

Methodology

Medieval II Total War has more than 600 individual units for 17 different playable factions. Some of them are common units that are just variations for different factions such as Peasants, most of siege engines, vessel units and some other infantry units and also some of them are very similar in terms of unit attributes. To understand unit balance in game, it's crucial to examine the effectiveness of the units that factions have in the battlefield. The game's battle engine creates a bunch of variables during the gameplay that affects the result of the battle. For example, if one side loses their general, their morale decreases dramatically and units with low morale become mentally vulnerable and tend to rout(leave the battlefield) much easier compared to a unit that has its general alive. There are many variables in the gameplay that trigger chain of events and that causes different and unique battle results on every play. That also creates a very complex situation to be able to do combat prediction.

We will evaluate the unit effectiveness by choosing a reference unit (Peasants of Byzantine Empire) and making all other units fight against Peasants. After each playtest session we will note the remaining soldiers of both armies and use that data to produce our combat prediction equation based on Lanchester's Attrition Laws for Medieval II Total War.

3.1 Units

Units are basically armies that have various amounts of soldiers. Soldier amounts of units change from 16 to 60. There are 9 unit types in Medieval II Total War.

3.1.1 Light infantry. Light infantry units are cheap and fast units in the game. They require a short time to recruit.

3.1.2 Spear infantry. Units with spears as primary weapons. Spear Infantry units are more effective against cavalry units but also less effective against sword units.

3.1.3 Heavy infantry. Units with more advanced armors and defense values. Heavy Infantry units are very effective on sieges when they are using ladders and siege towers.

3.1.4 Missile infantry. Missile Infantry units generally have bows and javelins as primary weapons. These units are very effective against Light Infantry and Light Cavalry units. Primary weapons have limited ammunition and when their ammunition finishes, they switch to their secondary weapons.

3.1.5 Light cavalry. Very useful units for chasing routing enemy units. Light Cavalry units are fast units with light armors.

3.1.6 Heavy cavalry. Slower units compared to Light Cavalry units. However they have more advanced armors. Heavy Cavalry units are very effective against any kind of infantry units.

3.1.7 Missile cavalry. Missile Cavalry units have a reflex to maintain distance with enemy units. They are very fast and mobile. They can make ranged attacks while running away from enemy units. However their secondary weapons are weaker among other units.

3.1.8 Siege engine. Siege Engine units are very effective against city walls. Even though they are designed to break city walls, they are also effective against any kind of land units with their extreme attack power.

3.1.9 Vessels. Vessels are sea units of the game. Vessel units have very large movement ranges but they are also non-playable units in Medieval II Total War.

3.2 Unit Stats

Unit stats are a projection of a unit’s abilities in the gameplay.

3.2.1 Infantry. Infantry represents the category that unit belongs to.

3.2.2 Morale. Morale represents the confidence that a unit has in victory. If a unit reaches "broken" morale, then it will rout. It is a mechanic present in all Total War games to date.

Unit's morale or else how easy it is for a unit to lose heart and flee the battle. The greater the value, the less likely the unit will rout. (The Complete EDU Guide, 2022)

3.2.3 Soldiers. Number of soldiers the unit has on gameplay.

3.2.4 Attack. Attack value of the unit's primary weapon (if the unit has missiles, those are its primary weapons).

3.2.5 Charge bonus. Increased melee attack and damage for 30 sec after charge. Units have to be far enough away to start a charge for it to count. Bonus counts against anything you run into while charging. Knockdown and penetration of the line is tied to speed and mass alone, while the stat bonus is simply that.

3.2.6 Range. Represents the range of missile infantry units that can make ranged attacks.

3.2.7 Ammunition. Ammunition that missile units ranged weapons has. They immediately switch to secondary weapons when they finish all their ammunition or collide with an enemy unit.

3.2.8 Total defense. Total Defense is the summary of Armor, Defense Skill and Shield.

3.2.9 Armour. Armor is essentially equivalent from all sides (with the caveat that follows) and has a disproportionate increased effect if base Armor value is 20 or more.

Second opinion is that armor works against ranged attacks.

3.2.10 Defense skill. Defense Skill applies fully to front only and gets a 33% penalty to left side defense, 20% penalty to right side defense and 50% penalty to back side defense (Where penalty is a percent of attack-defense skill value).

3.2.11 Shield. Shield applies fully to front and right (not front and left as it should be) and gets a 50% penalty (to Attack-Shield Defense) for back and left sides.

3.2.12 Hit points. Hit Points have been used on units since the early Total War games. According to players on twcenter.net, hit points represent the hits that unit soldiers can take before they die. All the units hit points value is between 1 and 2. That means the units that have 2 hit points need to take twice hits to lose a soldier.

3.2.13 Cost. Generation cost of the unit.

3.2.14 Upkeep. Upkeep cost is valid only in the campaign mode. Maintenance cost of the unit for 1 game turn.

3.2.15 Weapon upgrade. Upgrade cost of units weapons. Weapon upgrade affects attack values of the units.

3.2.16 Armor upgrade. Upgrade cost of units armor. Armor upgrade affects defense values of the units.

3.3 Factions

There are 22 different factions in Medieval II Total War. In the Grand Campaign mode, 17 factions are playable. However in the beginning only England, France,

Spain, The Holy Roman Empire and Venice are available to start. Other factions may be unlocked all at once when the player finishes the campaign mode by achieving the campaign rules told by the game in the beginning.

3.3.1 Starting positions of factions. Each faction has their spesific starting positions on the game map. Even tough this study aims to examine the unit balance in the battlefield, starting positions of some factions can be considered as and advantage in the campaign mode of the game.

3.3.1.1 The Byzantine Empire.

Figure 1. Starting position of The Byzantine Empire.

3.3.1.2 Denmark.

Figure 2. Starting position of Denmark.

3.3.1.3 Egypt.

Figure 3. Starting position of Egypt.

3.3.1.4 England.

Figure 4. Starting position of England.

3.3.1.5 France.

Figure 5. Starting position of France.

3.3.1.6 The Holy Roman Empire.

Figure 6. Starting position of The Holy Roman Empire.

3.3.1.7 Hungary.

Figure 7. Starting position of Hungary.

3.3.1.8 Milan.

Figure 8. Starting position of Milan.

3.3.1.9 Moors.

Figure 9. Starting position of Moors.

3.3.1.10 Poland.

Figure 10. Starting position of Poland.

3.3.1.11 Portugal.

Figure 11. Starting position of Portugal.

3.3.1.12 Russia.

Figure 12. Starting position of Russia.

3.3.1.13 Scotland.

Figure 13. Starting position of Scotland.

3.3.1.14 Sicily.

Figure 14. Starting position of Sicily.

3.3.1.15 Spain.

Figure 15. Starting position of Spain.

3.3.1.16 The Turks.

Figure 16. Starting position of The Turks.

3.3.1.17 Venice.

Figure 17. Starting position of Venice.

In the beginning of this research, all units and their stats provided by Creative Assembly have been noted on excel to have refined data and make analyses. Logic of the unit stats will be explained subsequently. The table shown below represents the total value of all unit's individual stats. According to data collected from totalwar.wiki, factions don't all have the same amount of units therefore total soldiers count and also total production cost that units have.

Table 1

Factions in Medieval II Total War and Sum of Faction Unit Attributes

Faction	Unit Count	Soldiers	Attack	Charge Bonus	Total Defence	Armour	Defence Skill	Shield	Hit Points	Cost
Byzantine Empire	27	1034	257	68	325	80	162	66	28	14240
Denmark	38	1378	370	116	473	157	201	81	42	22230
Egypt	32	1140	305	83	395	92	178	67	35	18750
England	39	1418	394	137	449	138	195	71	42	21130
France	43	1590	411	149	502	179	221	69	45	25880
The Holy Roman Empire	40	1442	386	125	459	154	199	73	42	23480
Hungary	38	1352	366	119	473	137	200	81	41	21490
Milan	41	1504	429	135	538	172	233	95	43	25080
Moors	36	1328	345	96	454	100	205	93	39	21800
Poland	36	1254	336	116	434	129	193	80	38	20570
Portugal	41	1520	429	118	497	147	240	77	43	24390
Russia	33	1168	328	87	378	99	182	61	35	19940
Scotland	33	1174	306	108	385	122	170	60	35	17890
Sicily	37	1400	378	115	487	145	220	89	39	22150
Spain	43	1600	437	123	524	153	249	89	45	25390
The Turks	35	1240	326	87	443	117	200	67	38	22730
Venice	41	1504	435	135	522	166	234	89	43	24860

According to data, some of the factions have more diversity in units. The Holy Roman Empire, Milan, Portugal, Spain and Venice factions have 40 or over units while the Byzantine Empire has 27 units in total which is the faction that has the lowest value in terms of unit count.

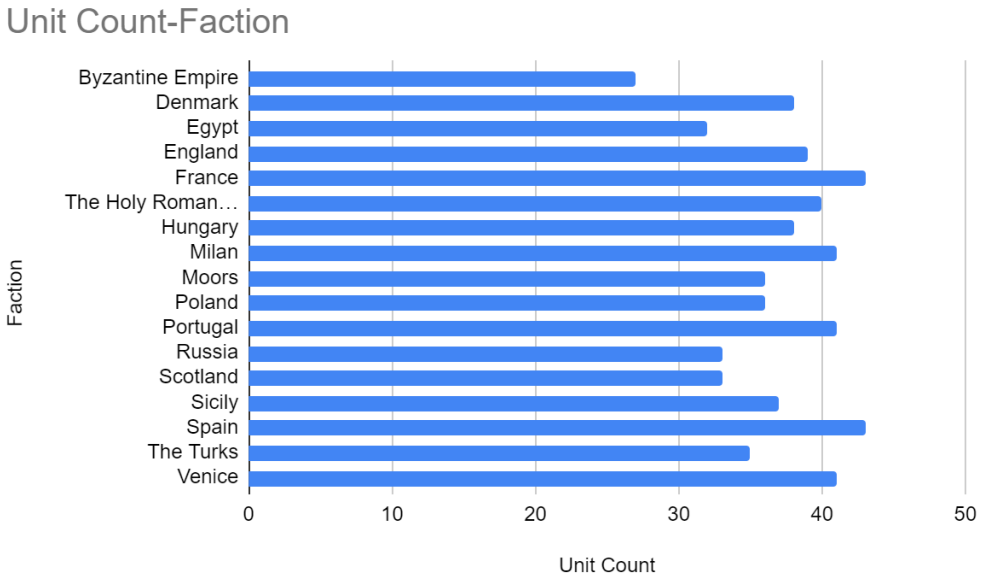

Figure 18. Unit Count of Factions.

Soldier count of units is variable. Units generally have 60, 48, 32 and 16 soldiers without the general. Soldier count changes according to unit count of the faction. If unit count is high, soldier count goes up relatedly. Total soldier count of all units of factions have shown on the table below. France, Portugal and Spain's soldier count of all their units is over 1500 and the factions with lowest soldier count are the Byzantine Empire, Egypt and Russia.

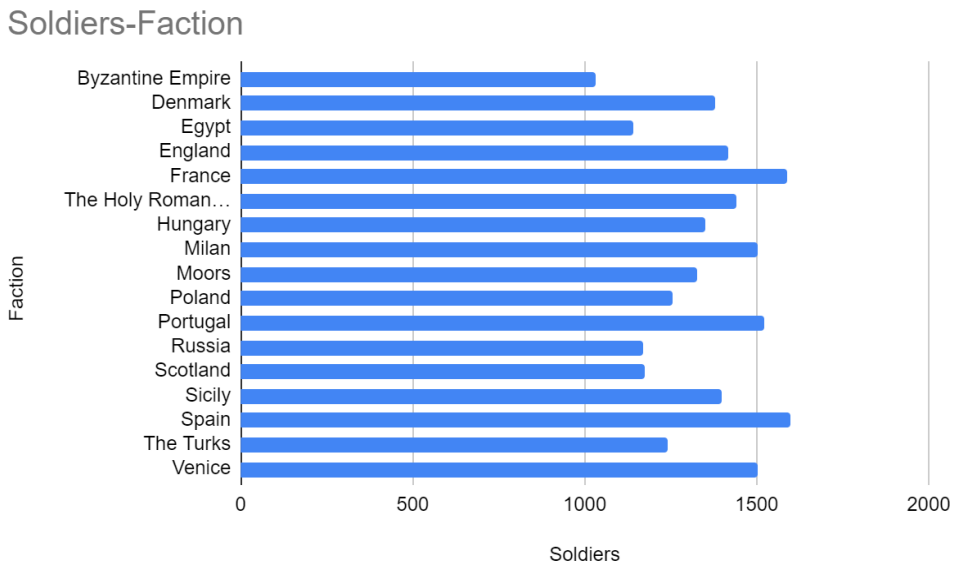

Figure 19. Soldier Count of Units of Each Faction.

Attack value of factions has been calculated by the sum of attack attributes of each unit that faction has. Attack values of factions are analogous with unit count value. Unit Count-Faction and Attack-Faction tables show that Unit Count and Attack values can be considered related.

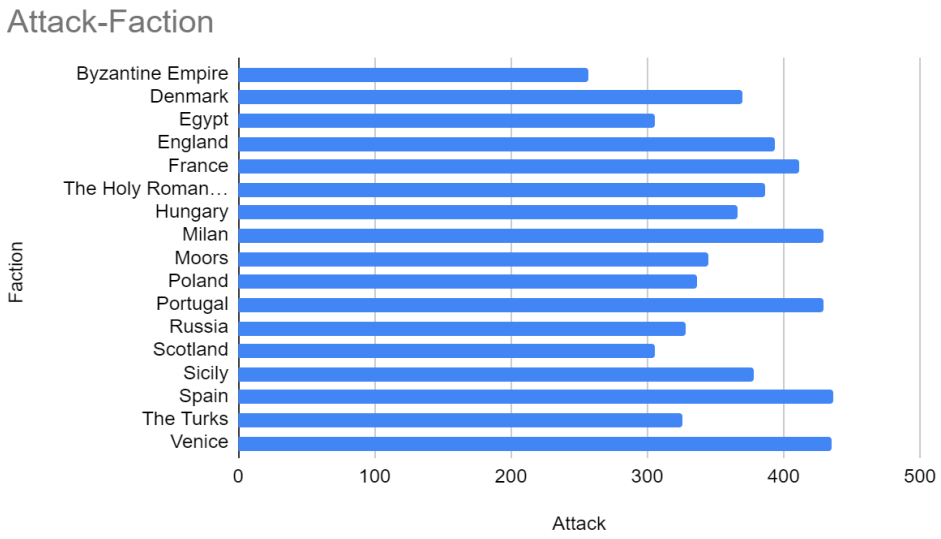

Figure 20. Sum of Attack Value of Units of Each Faction.

Total defense value of all units in factions are shown in the table below. Russia had a higher Attack value than Scotland. However in this table Scotland has higher Total Defense value than Russia.

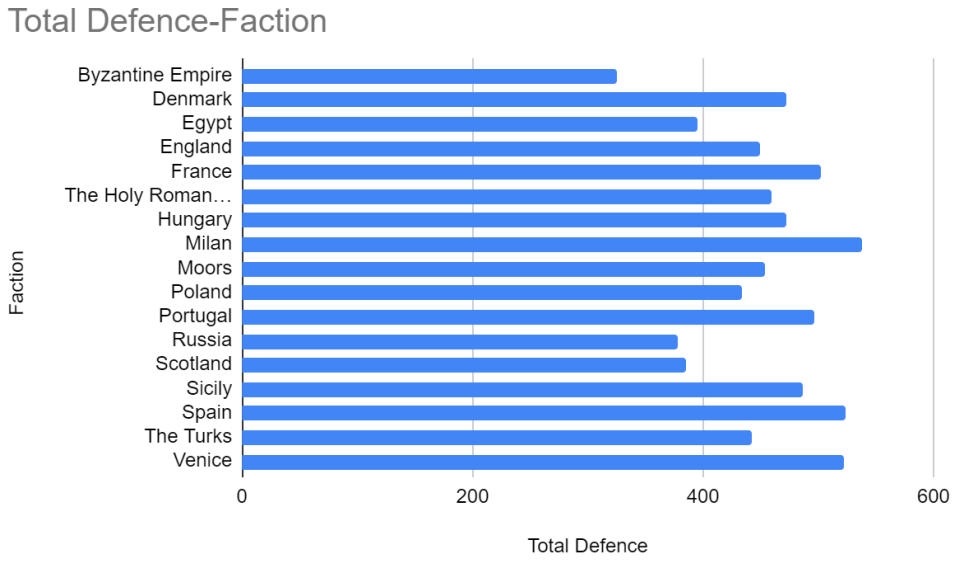

Figure 21. Sum of Total Defense Value of Units of Each Faction.

There are 8 factions with Defense Skill values over 200 which are Denmark, France, Milan, Moors, Portugal, Sicily, Spain and Venice.

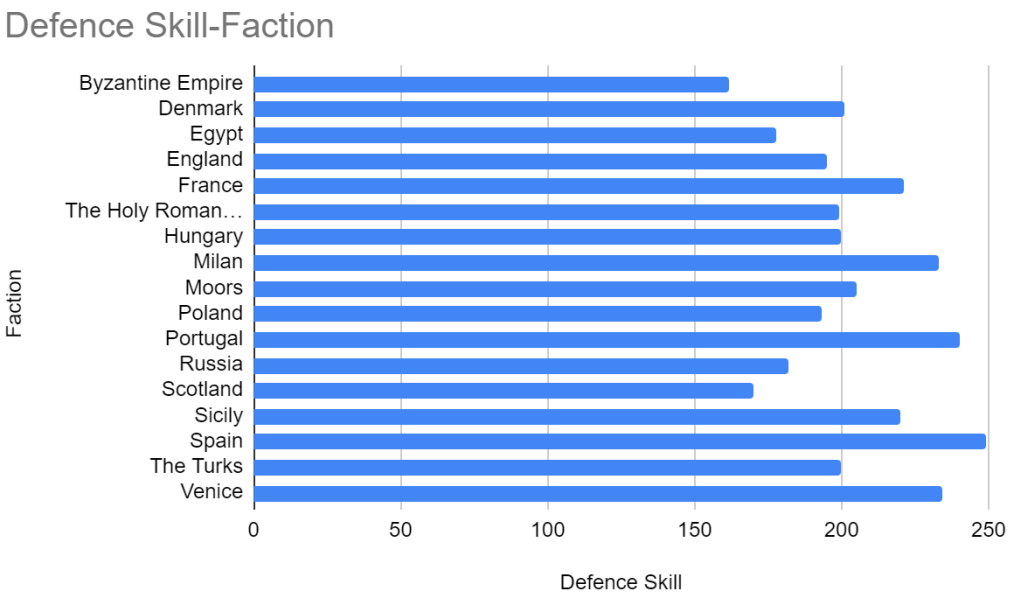

Figure 22. Sum of Defense Skill Value of Units of Each Faction.

The table below represents the cost of all units that factions have. Top 5 factions of this table are the same on the Unit Count-Faction table. Therefore Unit Count and Cost values are directly proportional.

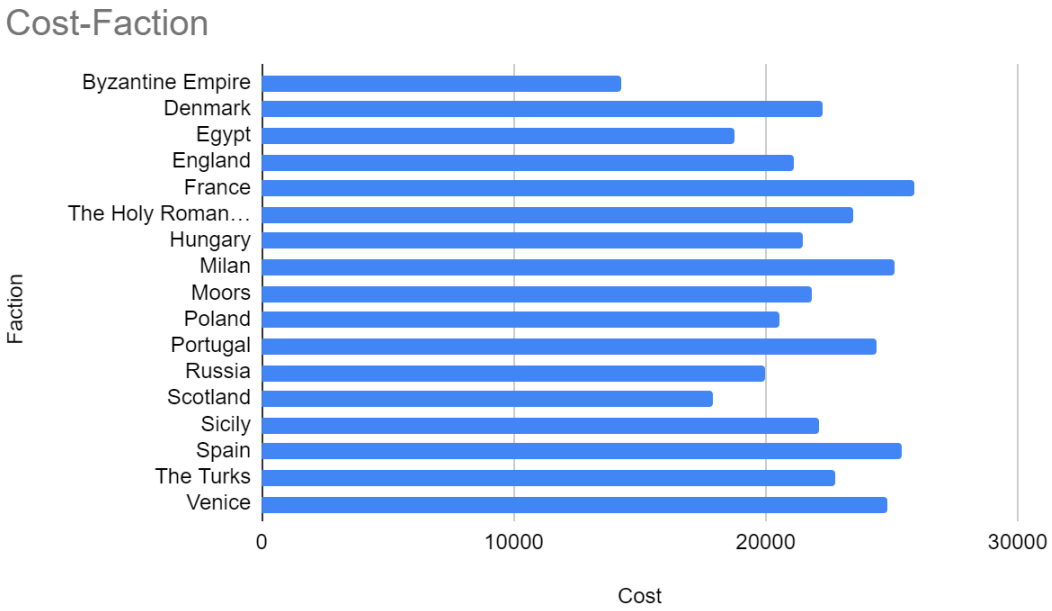

Figure 23. Sum of Unit Costs of Each Faction.

3.4 Playtest Environment

Medieval II Total War has a very complex battle engine and it had 6 years of development background and knowledge came from previous 3 Total War games when the game was published in 2006. The battle engine is capable of processing various variables that have an impact on the battle course such as morale, stamina, discipline, training level of the units and also impact of landforms and weather conditions among units.

We will use the game's battle engine to simulate how two individual units behave against each other and examine the battle results for strength and effectiveness comparison instead of building a simulation from scratch.

3.4.1 Game version and settings. We will make playtest sessions on game version 1.52 and unit scale will be set as “Small” on game settings.

3.4.2 Custom battle conditions. Game Type will be set to “Last Man Standing” and there will not be a time limit for all playtest sessions.

One of the most important settings in this phase is choosing a game map as “Grassy Plains”. Units on a higher ground have an advantage against units on lower ground in the battlefield. On that map there aren't any possible terrain difficulties such as hills or the opposite that might affect battle results. Therefore the Grassy Plains map will be our testbed for playtest sessions.

Figure 24. Settings for playtest sessions.

3.4.3 Unit selection screen. Team 1 is the unit that we control in the gameplay. Unit of the Team 1 will always be Peasants for all playtest sessions.

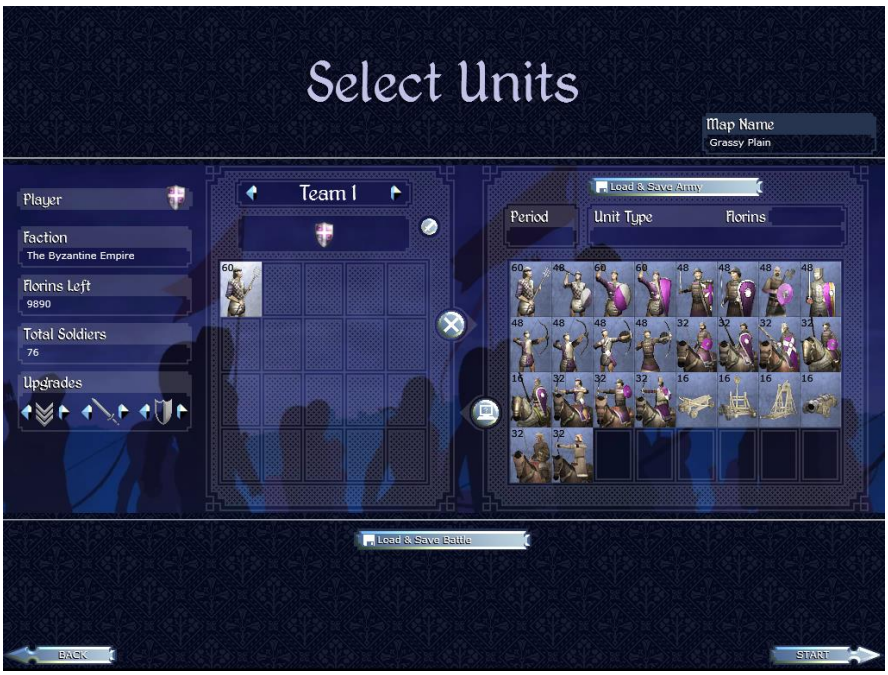

Figure 25. Unit selection screen of the controlled side.

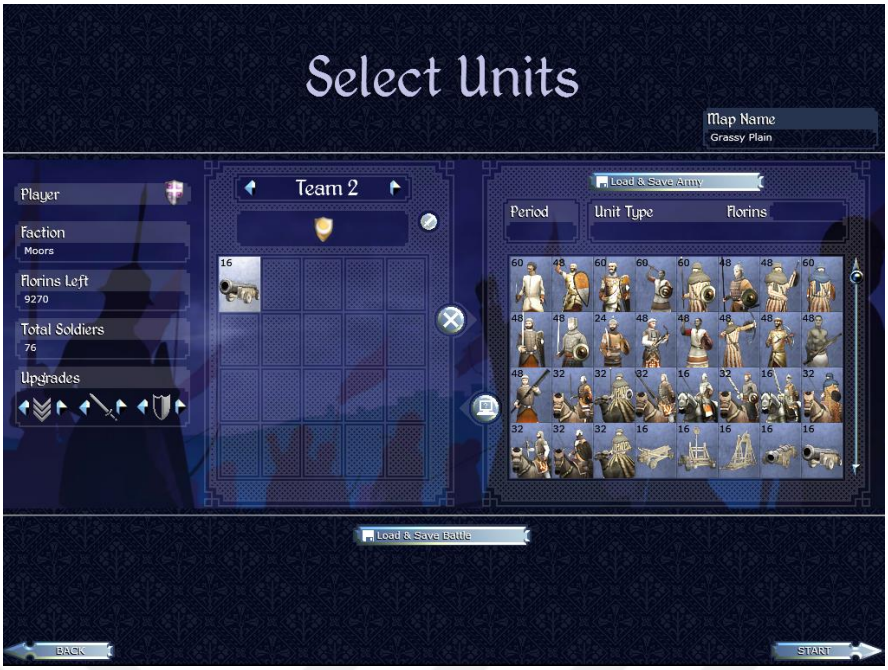

Figure 26. Unit selection screen of the opponent side.

3.5 Playtest Notes

When attacking, the controlled unit (Peasants of Byzantine Empire) has two speed variations while approaching the enemy unit. One is “Walk” and the other one is "March". When Walk mode is selected soldiers of the controlling unit approach the enemy by walking speed and when March mode is selected soldiers approach the enemy soldiers by running speed. In all playtests, controlling units have attacked the opponent unit in March mode to have a more fair battle against ranged units because controlling unit's soldiers would be lost before touching the enemy unit soldiers.

Ranged units have a reflex to maintain a distance to the opponent unit in the battlefield. If the melee enemy unit is close enough, the ranged unit starts to move away and try to reach range distance to be able to do a missile attack. If the moving speed of both units are similar, the ranged unit moves away from the enemy unit until it reaches the border of the battlefield. Eventually they must fight hand to hand because at the border of the battlefield, the ranged unit cannot maintain distance to the opponent melee unit.

Figure 27. Screenshot of the battlefield at the moment that ranged units kept their distance against melee units.

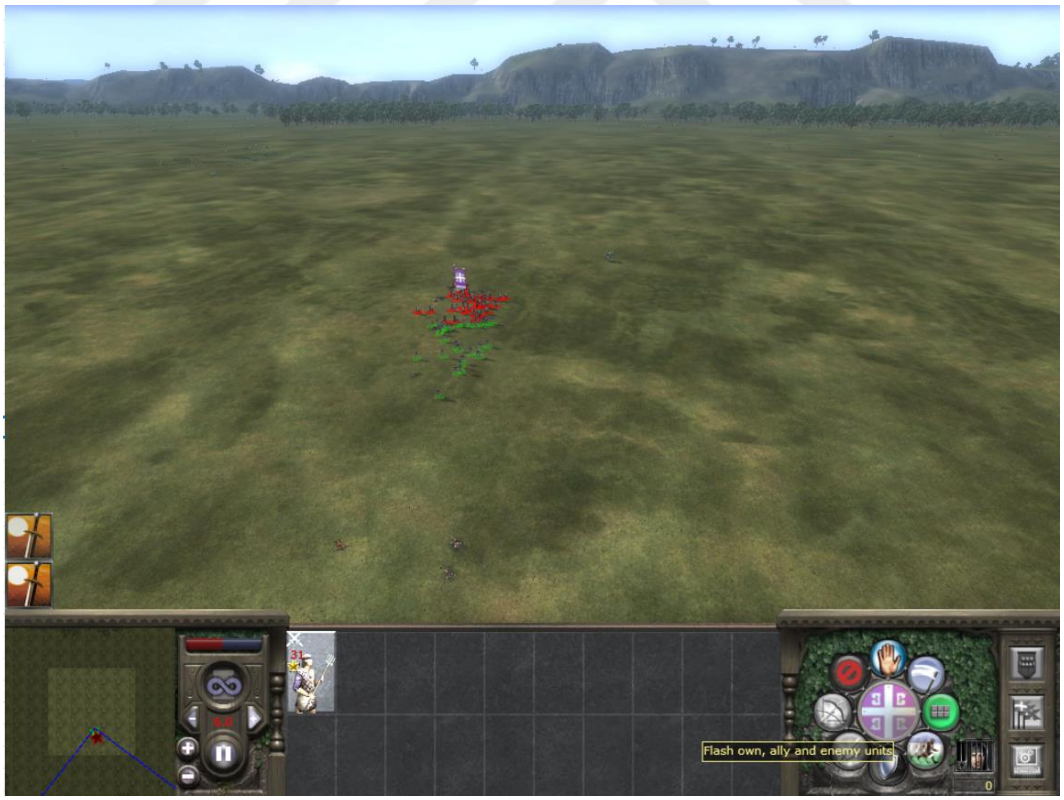

Figure 28. Ranged units switch to their secondary weapon at the border of the battlefield.

According to observations, ranged attacks of some siege engine units hit chance is very low against melee units and randomized. Ranged attacks of all the siege engine units could hit the Peasants before they reach the siege engine unit but hit rate of Bombard, Trebuchet and Cannon units is very low among others in all 3 individual playtests. All the siege engine units are designed to break walls but some of them like Bombard, Trebuchet and Cannon are specially designed for static units as walls so their ranged attacks are not really efficient against melee units.

Almost all the siege engines could make ranged attacks at least once before the enemy unit reached. Unlike missile units like archers and crossbowmens, siege engines don't have a reflex to maintain distance to be able to do more ranged attacks before the enemy unit reaches it, so they start to fight hand to hand immediately after the collision.

In battles between Peasants vs Archer units, Peasants units losts 20%-25% of their mens before collision. However Peasants win rate is still very high especially among early Archer units even if the soldier count of the both units are same or similar at the collision moment.

Figure 29. Battle results of Peasants vs. Peasants Crossbowman of France Faction.

Cavalry units are very effective against controlled units(Peasants of Byzantine Empire). Peasants lost 65%-70% of their men at the collision moment and their general dies very often by the collision. After a few seconds of the collision, Peasant unit soldiers start routing with white flag and the battle ends after a few seconds immediately.

Figure 30. Screenshot at the collision moment of cavalry and infantry units.

Apparently, peasant units are much more than simple early era units. They are cheap and effective on the battlefield even against much powerful enemies.

Hungarian basilisk is the only unit won against peasants in at least one session. After a missile attack peasants have lost ⅓ of their men and started to rout even if the general is alive.

Figure 31. Battle statistics of the first playtest session of Peasants vs. Bombard of Moors.

3.6 Reading Playtest Data

To understand which faction in the game is rich or the opposite in terms of unit varieties, we need to sort all the units in the game by effectiveness and making them fight against each other and observe the results is a good way to do that.

Peasants of the Byzantine Empire is our test unit that we will make the rest of the units of the game will fight against. The reason is that the Byzantine Empire is the first faction on the list at totalwar.wiki which is the website all the unit attribute datas is collected from and Peasants unit of Byzantine Empire is the first unit of the faction. Another reason is that the Peasants unit is one of the most common units for all factions and it's also a basic unit in terms of its unit attributes that will not cause long playtest sessions.

All the units in the game will be fought 3 times with Peasants of Byzantine Empire and after every playtest session remaining soldiers of both sides will be noted.

Table 2

Playtest Results of the First 7 Units of Denmark Against Controlled Unit

Opponent Unit Faction	Opponent Unit	Battle 1 Results		Battle 2 Results		Battle 3 Results	
		Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining
Denmark	Peasants	40	35	39	32	37	30
Denmark	Town Militia	17	44	20	33	37	17
Denmark	Viking Raiders	18	56	26	51	15	52
Denmark	Spear Militia	28	15	34	17	39	28
Denmark	Dismounted Feudal Knights	8	49	9	47	7	49
Denmark	Dismounted Huscarls	26	46	15	47	4	46
Denmark	Norse Axemen	7	43	6	42	30	44

Playtest sessions have been made as 1v1 and sessions ended immediately when one of the sides started to rout and leave the battlefield. As seen above some playtest session results are not consistent and change dramatically in every session. According to observations during the playtest sessions the general is playing a crucial role on units morale and eagerness to fight against opponent units on the battlefield. Units tend to rout in very short time right after their general dies if units morale value is not very high. Remaining soldiers change in every playtest session consequently because the general dies at different times in every session.

Figure 32. Session ended immediately when one of the sides started to rout.

As mentioned before, all the units have been faced off with Peasants unit of Byzantine Empire for the playtest sessions and Peasants unit have been acknowledged as reference for evaluating all units efficiency. However another reference unit needed to evaluate the efficiency of Byzantine Empire units because setting a playtest environment for units belonging to the same faction is impracticable. To exceed this issue we look at the playtest results of other versions of Peasants of different factions. Evidently remaining soldiers of the Byzantine Empire and the Milan's Peasants are the closest. Therefore Peasants of Milan will be considered as a reference unit for the playtests against Byzantine Empire units.

Table 3

Playtest results of Peasants of Each Faction

		Battle 1 Results		Battle 2 Results		Battle 3 Results		Average		Dead Out of Total		Percentage of Dead	
Byzantine Empire	Peasants	21	28	14	30	23	41	19.33	33.00	41.67	28.00	68.31%	45.90%
Denmark	Peasants	40	35	39	32	37	30	38.67	32.33	22.33	28.67	36.61%	46.99%
Egypt	Peasants	43	23	43	12	37	17	41.00	17.33	20.00	43.67	32.79%	71.58%
England	Peasants	25	25	23	41	19	42	22.33	36.00	38.67	25.00	63.39%	40.98%
France	Peasants	20	33	22	47	29	26	23.67	35.33	37.33	25.67	61.20%	42.08%
The Holy Roman Empire	Peasants	39	20	24	30	13	43	25.33	31.00	35.67	30.00	58.47%	49.18%
Hungary	Peasants	20	30	37	32	18	35	25.00	32.33	36.00	28.67	59.02%	46.99%
Milan	Peasants	21	41	26	11	22	21	23.00	24.33	38.00	36.67	62.30%	60.11%
Moors	Peasants	35	9	45	28	32	24	37.33	20.33	23.67	40.67	38.80%	66.67%
Poland	Peasants	24	23	39	15	30	29	31.00	22.33	30.00	38.67	49.18%	63.39%
Portugal	Peasants	35	28	44	41	47	38	42.00	35.67	19.00	25.33	31.15%	41.53%
Russia	Peasants	32	24	14	35	34	30	26.67	29.67	34.33	31.33	56.28%	51.37%
Sicily	Peasants	39	27	43	25	29	11	37.00	21.00	24.00	40.00	39.34%	65.57%
Spain	Peasants	48	29	35	19	11	33	31.33	27.00	29.67	34.00	48.63%	55.74%
The Turks	Peasants	44	24	35	24	36	27	38.33	25.00	22.67	36.00	37.16%	59.02%
Venice	Peasants	40	27	19	33	36	27	31.67	29.00	29.33	32.00	48.09%	52.46%

When all the units have fought 3 times against Peasants of Byzantine Empire, the average of remaining soldiers of both sides in 3 playtest sessions has been calculated.

Table 4

Representation Average Remaining Soldiers

			Battle 1 Results		Battle 2 Results		Battle 3 Results		Average	
2	Opponent Unit Faction	Opponent Unit	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining
29	Denmark	Spear Militia	28	15	34	17	39	28	33,66666667	20
30	Denmark	Dismounted Feudal Knights	8	49	9	47	7	49	8	48,33333333
31	Denmark	Dismounted Huscarls	26	46	15	47	4	46	15	46,33333333
32	Denmark	Norse Axemen	7	43	6	42	30	44	14,33333333	43
33	Denmark	Norse Swordsmen	16	45	17	42	10	47	14,33333333	44,66666667
34	Denmark	Obudshaer	18	41	14	46	0	37	10,66666667	41,33333333
35	Denmark	Sword Staff Militia	1	23	15	40	12	40	9,333333333	34,33333333

With the average of the 3 individual playtest session results, we could get the number of dead soldiers for both sides. As told above, all the attributes of the units have been collected on totalwar.wiki. Soldier count of the units on totalwar.wiki has shown without the general of the unit. In the gameplay, all the units are seen as soldier count value on totalwar.wiki +1 (the general).

Table 5

Representation of Dead Count out of Total Soldier Count

Opponent Unit Faction	Opponent Unit	Battle 1 Results		Battle 2 Results		Battle 3 Results		Average		Dead Out of Total	
		Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test	Opponent
Denmark	Spear Militia	28	15	34	17	39	28	33,66666667	20	27,33333333	41
Denmark	Dismounted Feudal Knights	8	49	9	47	7	49	8	48,33333333	53	0,666666667
Denmark	Dismounted Huscarls	26	46	15	47	4	46	15	46,33333333	46	2,666666667
Denmark	Norse Axemen	7	43	6	42	30	44	14,33333333	43	46,66666667	6
Denmark	Norse Swordsmen	16	45	17	42	10	47	14,33333333	44,66666667	46,66666667	4,333333333
Denmark	Obudshaer	18	41	14	46	0	37	10,66666667	41,33333333	50,33333333	7,666666667
Denmark	Sword Staff Militia	1	23	15	40	12	40	9,333333333	34,33333333	51,66666667	14,66666667

With the data of dead soldiers counted, we can now calculate the percentage of dead for both sides with the formula of "(Soldier Count - Average Remaining) / Soldier Count"

Table 6

Representation of Percentage of Dead out of Total Soldier Count

Opponent Unit Faction	Opponent Unit	Battle 1 Results		Battle 2 Results		Battle 3 Results		Average		Dead Out of Total		Percentage of Dead	
		Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test Remaining	Opponent Remaining	Test	Opponent	Test	Opponent
Denmark	Spear Militia	28	15	34	17	39	28	33,66666667	20	27,33333333	41	44,81%	67,21%
Denmark	Dismounted Feudal Knights	8	49	9	47	7	49	8	48,33333333	53	0,666666667	86,89%	1,36%
Denmark	Dismounted Huscarls	26	46	15	47	4	46	15	46,33333333	46	2,666666667	75,41%	5,44%
Denmark	Norse Axemen	7	43	6	42	30	44	14,33333333	43	46,66666667	6	76,50%	12,24%
Denmark	Norse Swordsmen	16	45	17	42	10	47	14,33333333	44,66666667	46,66666667	4,333333333	76,50%	8,84%
Denmark	Obudshaer	18	41	14	46	0	37	10,66666667	41,33333333	50,33333333	7,666666667	82,51%	15,65%
Denmark	Sword Staff Militia	1	23	15	40	12	40	9,333333333	34,33333333	51,66666667	14,66666667	84,70%	29,93%
Denmark	Arquebusiers	40	17	32	17	21	36	31	23,33333333	30	25,66666667	49,18%	52,38%
Denmark	Crossbowmen	15	37	10	34	14	43	13	38	48	11	78,69%	22,45%
Denmark	Crossbow Militia	28	7	10	15	28	21	22	14,33333333	39	34,66666667	63,93%	70,75%

Percentage of dead soldiers will help us to give units ranking. The logic of that is on the line the highest percentage of dead value for the test unit which is Peasants of Byzantine Empire and the lowest percentage of dead value for the opponent unit will give us the most effective unit in the game because all the units have been faced off with Peasants in exact same conditions with the same chain of interventions. Even if the results depend on many things, most importantly the dying moment of the general, it is one way to evaluate the effectiveness of a game unit on the battlefield. However, this method gives healthy results only at infantry vs. infantry matchings. Therefore, while the ranking results of cavalry units are meaningful among the cavalry units category, it will be opposite among units of other unit categories.

3.7 Combat Prediction with Lanchester's Attrition Laws

Lanchester's laws are mathematical formulae for calculating the relative strengths of military forces. The Lanchester equations are differential equations describing the time dependence of two armies' strengths A and B as a function of time, with the function depending only on A and B (Lanchester, 1916).

3.7.1 Example equations. Suppose that two armies, Red and Blue, are engaging each other in combat. Red is shooting a continuous stream of bullets at Blue. Meanwhile, Blue is shooting a continuous stream of bullets at Red.

Let symbol A represent the number of soldiers in the Red force. Each one has offensive firepower α, which is the number of enemy soldiers it can incapacitate (e.g., kill or injure) per unit time. Likewise, Blue has B soldiers, each with offensive firepower β.

Lanchester's square law calculates the number of soldiers lost on each side using the following pair of equations(Taylor, 1983).

Here, dA/dt represents the rate at which the number of Red soldiers is changing at a particular instant. A negative value indicates the loss of soldiers. Similarly, dB/dt represents the rate of change of the number of Blue soldiers.

$$\frac{\mathrm{d}A}{\mathrm{d}t} = -\beta B$$
$$\frac{\mathrm{d}B}{\mathrm{d}t} = -\alpha A$$

3.8 Using Lanchester's Attrition Laws in Medieval II Total War

Lanchester's Attrition Laws were used for combat prediction in World War I. We will use Lanchester's Attrition Laws to predict combat outcomes in the Medieval II Total War game using playtest results.

Playtest data has been created to avoid difficulty of processing abundant unit attributes. There isn't an official explanation of how unit attributes work and interact with each other. All the definitions of those attributes have been collected from inveterate player communities. All the definitions have been made for the Rome Total War game but according to modder players, there aren't major differences between Rome Total War and Medieval II Total War battle engines. We will combine the Lanchester's Attrition Laws and some unit attributes to create a formula for simulating Medieval II Total War's battle engine and making combat predictions.

In the example equations given above, "α" and "β" will represent the firepower of both armies in our combat prediction equation.

Lanchester equations shows that:

- If $\alpha=\beta$, i.e. the two sides have equal firepower, the side with more soldiers at the beginning of the battle will win;
- If $A=B$, i.e. the two sides have equal numbers of soldiers, the side with greater firepower will win;
- If $A>B$ and $\alpha>\beta$, then Red will win, while if $A<B$ and $\alpha<\beta$, Blue will win;
- If $A>B$ but $\alpha<\beta$, or $A<B$ but $\alpha>\beta$, the winning side will depend on whether the ratio of β/α is greater or less than the square of the ratio of A/B. Thus, if numbers and firepower are unequal in opposite directions, a superiority in firepower equal to the square of the inferiority in numbers is required for victory; or, to put it another way, the effectiveness of the army rises proportionate to the square of the number of people in it, but only linearly with their fighting ability.

We will estimate the firepower of both sides by using playtest data and implementing that data into our equation.

3.8.1 Battle example. Imagine Peasants vs Spear Militia match. According to base unit stats analysis, we expect Spear Militia to win the battle. However, playtest results show that win rates of Peasants are higher among Spear Militia.

When we have a closer look at the stats of both units, Peasants have 4 Attack value and 3 Total Defense value while Spear Militia have 5 Attack value and 7 Total Defense value. However, the Defense Skill value of Peasants are higher than the Spear Militia. In this case, reflection of those stats on the battlefield will be like below;

- Peasants attack: 4-1 = 3 chance to wound
- Spear Militia attack: 5-3 = 2 chance to wound.

So we see that the peasants will be winning the battle, which also means their morale stays high.

We will test our combat prediction equation on Peasants(Byzantine Empire) vs Spear Militia(Denmark) match also.

Figure 33. Unit stats of Peasants and Spear Militia.

As seen above, Peasants have 4 Attack and 3 Defense Skill values while Spear Militia have 5 Attack and 1 Defense Skill values.

For our combat prediction equation, we consider:

α = Unit 1 Damage

β = Unit 2 Damage

α = Unit 1 Attack Value - Unit 2 Defense Skill Value

β = Unit 2 Attack Value - Unit 1 Defense Skill Value

In this case:

- Damage value of Peasants: 4-1 =3
- Damage value of Spear Militia: 5-2 = 2

Table 7

Battle Results of Peasants Against Spear Militia Units

Faction	Rank	Unit	Unit Type	Battle 1 Results		Battle 2 Results		Battle 3 Results	
Byzantine Empire	324	Spear Milita	Spear	18	35	14	35	18	26
Hungary	351	Spear Militia	Light	30	12	15	41	25	37
Egypt	367	Spear Militia	Spear	6	29	29	6	16	24
Russia	370	Spear Militia	Light	28	26	33	14	9	33
Portugal	376	Spear Militia	Spear	8	28	41	25	24	16
The Turks	380	Spear Militia	Spear	10	35	41	24	37	22
Poland	382	Spear Militia	Light	23	16	20	35	27	10
England	405	Spear Militia	Spear	26	8	35	10	13	28
Denmark	416	Spear Militia	Light	28	15	34	17	39	28
Moors	418	Spear Militia	Spear	46	26	49	27	26	24
Scotland	420	Spear Militia	Spear	32	23	36	28	33	3
France	423	Spear Militia	Spear	41	24	35	12	30	21
Spain	425	Spear Militia	Spear	24	11	30	7	27	10
The Holy Roman Empire	430	Spear Militia	Spear	34	18	45	10	35	27

As seen above, the winning rate of Peasants is quite high surprisingly even though their attack value is lower than the Spear Militia. This means the equation we have declared for unit damages makes sense.

As we mentioned before, the general of the unit is playing a crucial role on unit morale and units that lost their general tend to rout. The general's death is dependent on various events. General might be on the front line and may die earlier than we expect and that causes the unit to rout or victory for the unit that lost its general could be a distinct possibility if they have more men remaining than the opponent unit. In that case the unit may decide to fight without their general and even could win the battle. In short, the victory is randomized by various variables and having more damage or defense value than the opponent unit does not mean victory. Therefore we need to implement a randomized factor onto our combat prediction equation to give the units with lower attack or defense value a possibility of winning as we see on battle results. In order to do that, we will tell our equation to pick a value for the damage of a unit between α and 1.17^{α}.

Unit 1 and Unit 2 damage values will be:

$\alpha = \alpha < x < 1.17^{\alpha}$

$\beta = \beta < x < 1.17^{\beta}$

Representation of the equation above in Python will be:

“Unit1Damage=random.uniform(Unit1Attack-Unit2Defence,((1.17**(Unit1Attack-Unit2Defence)) * Unit2Attack-Unit1Defence))

Unit2Damage=random.uniform(Unit2Attack-Unit1Defence,((1.17**(Unit2Attack-Unit1Defence)) * Unit1Attack-Unit2Defence))”

“α” and “β” are the total values of the soldier count in a unit. Opponent soldier counts will be decreased up to the value of damage in every sequence and their damage value will be updated instantly with new soldier count as expected. For example, if a unit has 100 soldiers and the damage value of the unit is 10, when the unit loses 50 soldiers its damage value will be 5.

```
Unit 1 HP:  61 Unit 1 Damage:  3.31
Unit 2 HP:  61 Unit 2 Damage 3.99
Unit 1 HP:  57.01 Unit 1 Damage:  3.09
Unit 2 HP:  57.69 Unit 2 Damage:  3.77
Unit 1 HP:  53.25 Unit 1 Damage:  2.89
Unit 2 HP:  54.59 Unit 2 Damage:  3.57
Unit 1 HP:  49.68 Unit 1 Damage:  2.7
Unit 2 HP:  51.7 Unit 2 Damage:  3.38
Unit 1 HP:  46.3 Unit 1 Damage:  2.51
Unit 2 HP:  49.01 Unit 2 Damage:  3.2
Unit 1 HP:  43.1 Unit 1 Damage:  2.34
Unit 2 HP:  46.49 Unit 2 Damage:  3.04
Unit 1 HP:  40.06 Unit 1 Damage:  2.17
Unit 2 HP:  44.15 Unit 2 Damage:  2.88
```

Figure 34. Results of the combat prediction tool.

As seen above both units have the same amount of soldiers (Unit 1 HP and Unit 2 HP) and their damage values are different. Each line is an individual sequence and both units HP and Damage values change each sequence. Unit 1 has 61 soldiers and 3.31 damage value in the beginning but when it has 40.06 soldiers remaining, its damage value is 2.17.

In order to update HP and Damage values each sequence, following equation should be used:

α = Unit 1
β = Unit 2

$\alpha_{HP} = \alpha_{Hp} - \beta_{damage}$
$\alpha_{damage} = (\alpha_{hp} \times \alpha_{damage}) / \alpha_{Old\ HP}$

$\beta_{HP} = \beta_{Hp} - \alpha_{damage}$
$\beta_{damage} = (\beta_{hp} \times \beta_{damage}) / \beta_{Old\ HP}$

Representation of the equation above in Python will be:

```
Unit1OldHP = Unit1HP
Unit2OldHP = Unit2HP

Unit1HP = Unit1HP - Unit2Damage;
Unit2HP = Unit2HP - Unit1Damage;
Unit1Damage = Unit1HP * Unit1Damage / Unit1OldHP;
Unit2Damage = Unit2HP * Unit2Damage / Unit2OldHP;

Unit1OldHP = Unit1HP
Unit2OldHP = Unit2HP
```

According to unit stats of Peasants and Spear Militia, Peasants damage value is 3 and Spear Militia damage value is 2. Also in the battle results we see that the Spear Militia unit have won the battle against Peasants 4 times even if its damage value is lower than the Peasants. As we mentioned before if a unit loses its general, their morale decreases dramatically and they start to rout eventually. This is one of the reasons units with lower damage value may win the battle.

```
Unit 1 HP:  61 Unit 1 Damage:  3.01
Unit 2 HP:  61 Unit 2 Damage:  3.49
Unit 1 HP:  57.51 Unit 1 Damage:  2.84
Unit 2 HP:  57.99 Unit 2 Damage:  3.32
Unit 1 HP:  54.19 Unit 1 Damage:  2.68
Unit 2 HP:  55.14 Unit 2 Damage:  3.15
Unit 1 HP:  51.04 Unit 1 Damage:  2.52
Unit 2 HP:  52.47 Unit 2 Damage:  3.0
Unit 1 HP:  48.04 Unit 1 Damage:  2.37
Unit 2 HP:  49.94 Unit 2 Damage:  2.86
Unit 1 HP:  45.18 Unit 1 Damage:  2.23
Unit 2 HP:  47.57 Unit 2 Damage:  2.72
Unit 1 HP:  42.46 Unit 1 Damage:  2.1
Unit 2 HP:  45.34 Unit 2 Damage:  2.59
```

Figure 35. Prediction tool results for the matching of Peasants vs. Spear Militia.

When we re-run the same code, damage values of both units change. As seen above, the damage value of Unit 2 may be higher than Unit 1 and Unit 2 may win the battle. We set the damage value of both units with "$\alpha = \alpha < x < 1.17^{\alpha}$" and "$\beta = \beta < x < 1.17^{\beta}$" equations so the damage values of both units change every time the code is run. That way units with closer damage values may win against each other.

3.9 Combat Prediction Tool

```
import random
Unit1HP= 61
Unit2HP= 61
Unit1Attack=4
Unit2Attack=5
Unit1Defense=3
Unit2Defense=1
Unit1Damage= random.uniform(Unit1Attack-Unit2Defense,((1.17**(Unit1Attack-Unit2Defense))* Unit2Attack-Unit1Defense))
Unit2Damage= random.uniform(Unit2Attack-Unit1Defense,((1.17**(Unit2Attack-Unit1Defense))* Unit1Attack-Unit2Defense))

Unit1OldHP = Unit1HP
Unit2OldHP = Unit2HP
print("Unit 1 HP: ", Unit1HP, "Unit 1 Damage: ", round(Unit1Damage,2))
print("Unit 2 HP: ", Unit2HP, "Unit 2 Damage: ", round(Unit2Damage,2))
while Unit1HP >=0 and Unit2HP >=0:
   Unit1HP = Unit1HP - Unit2Damage;
   Unit2HP = Unit2HP - Unit1Damage;
   Unit1Damage = Unit1HP * Unit1Damage / Unit1OldHP;
   Unit2Damage = Unit2HP * Unit2Damage / Unit2OldHP;
   Unit1OldHP = Unit1HP
   Unit2OldHP = Unit2HP

        print("Unit 1 HP: ",round(Unit1HP,2), "Unit 1 Damage: ",round(Unit1Damage,2))
        print("Unit 2 HP: ",round(Unit2HP,2), "Unit 2 Damage: ",round(Unit2Damage,2))
```

Chapter 4

Findings

Findings linked to playtest results and unit rankings are presented in this chapter.

4.1 Top 20 Infantry Units

While interpreting the data collected with playtests, we need to understand that this data shows the effectiveness of the units on the battlefield against only one unit which is Peasants of Byzantine Empire and having a lot of units that are in the Top 20 wouldn't make that country the strongest playable faction because the data we collected came from the tactical layer of the game only. However, there is a very deep and complex strategic layer of the game also when players have so many neighbors and hostiles around them, making diplomatic relations, gathering and using resources, having convenient lands and many others.

With the method explained above, we can get the ranking of the Infantry units sorted according to effectiveness on the gameplay. As seen on the graph below, Spain has the most units that are in the Top 20.

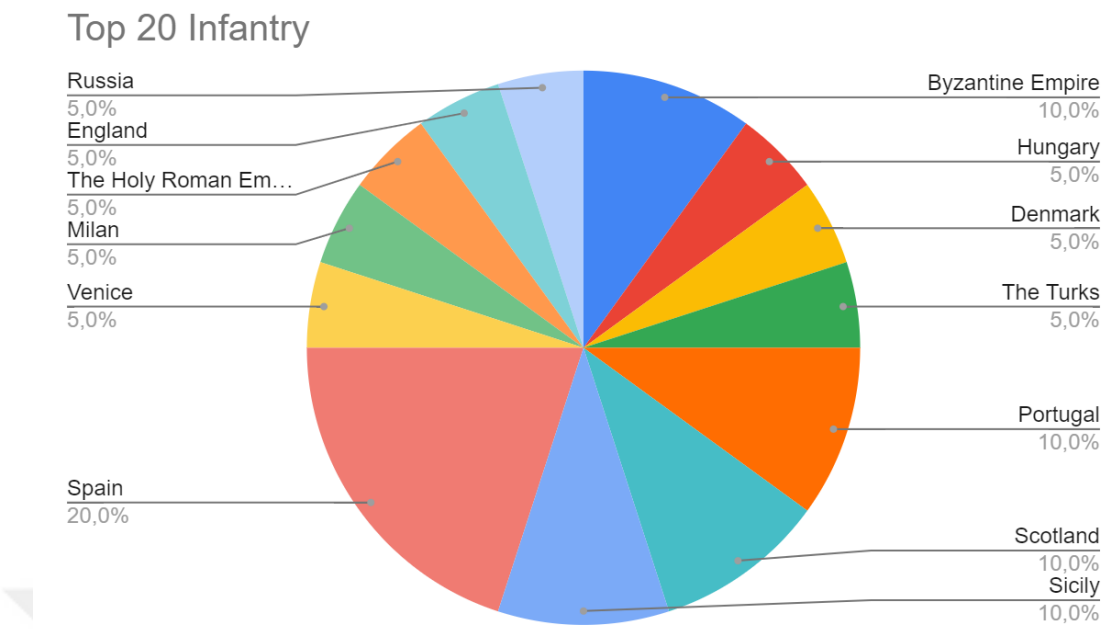

Figure 36. Distribution of Top 20 Infantry Units to Factions.

Russia's %5 of their units on the Top 20 is understandable at some point because Russia is one of the factions that has the least neighbors according to its geopolitical location in the game. Also, Moors don't have any units in the Top 20 infantry units. As we consider Moors to have only one natural enemy in the beginning of the game, the same evaluation can be made for Moors faction.

Figure 37. Moors starting location on the game map.

4.2 Top 20 Cavalry Units

Top 20 Cavalry units are also Top 20 units of the game for all unit categories therefore cavalry units are very effective against infantry units especially our controlling unit Peasants.

France and Poland are richer than other factions about effective cavalry units.

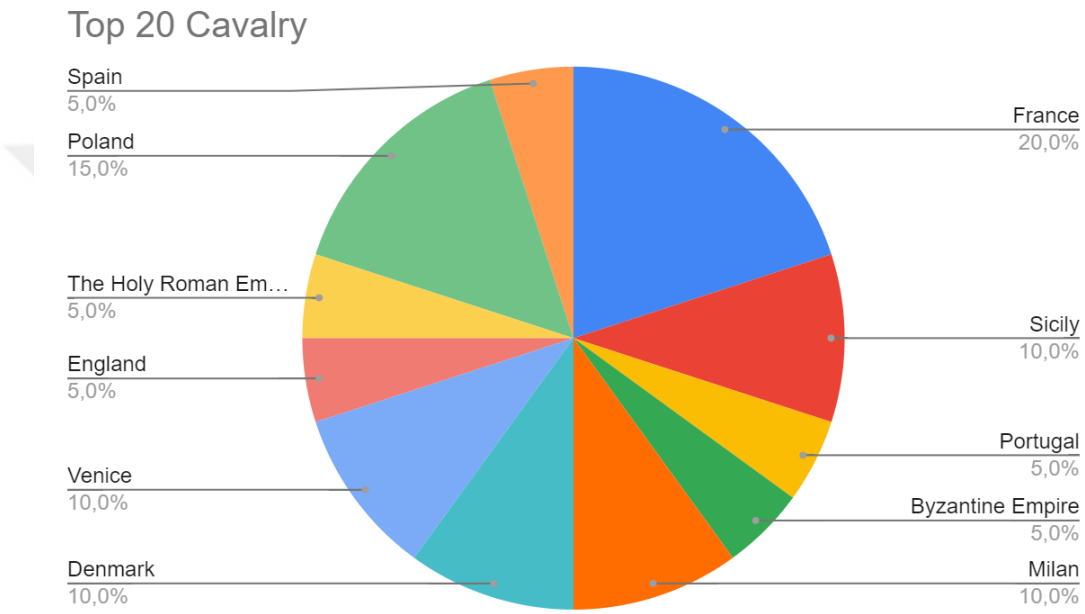

Figure 38. Distribution of Top 20 Cavalry Units to Factions.

4.3 Top 20 Light Infantry Units

Hungary and Spain draw attention on this graph.

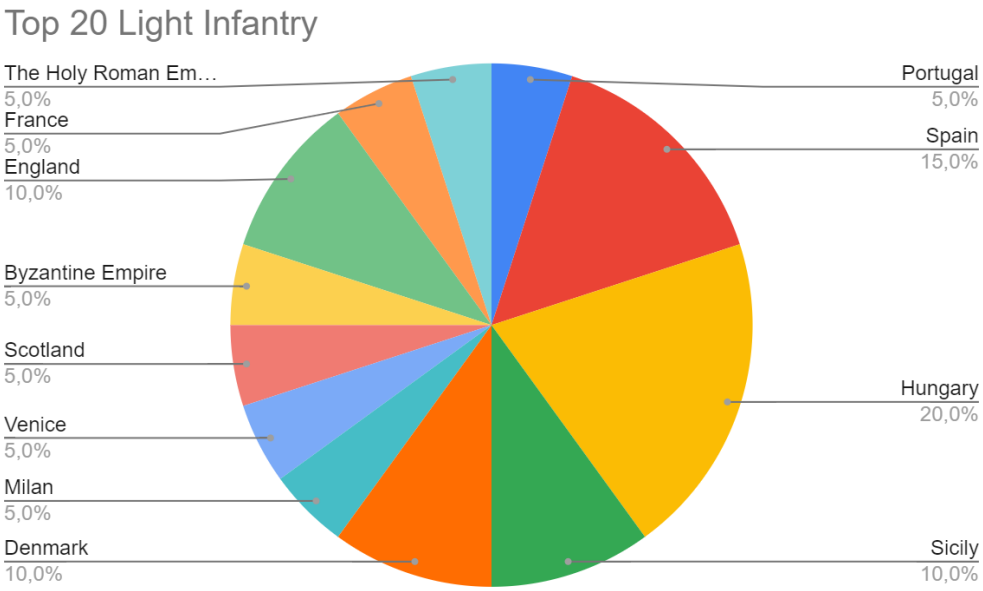

Figure 39. Distribution of Top 20 Light Infantry Units to Factions.

4.4 Top 20 Heavy Infantry Units

All the factions are substantially similar to the Heavy Infantry category.

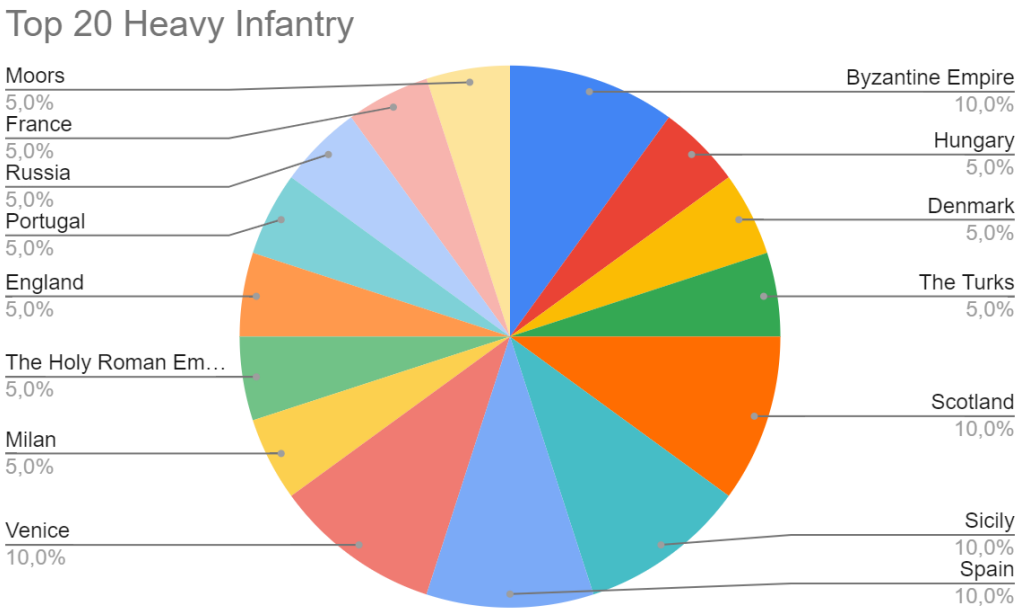

Figure 40. Distribution of Top 20 Heavy Infantry Units to Factions.

4.5 Top 20 Spear Infantry Units

Venice has 3 units in the Top 20 Spear Infantry.

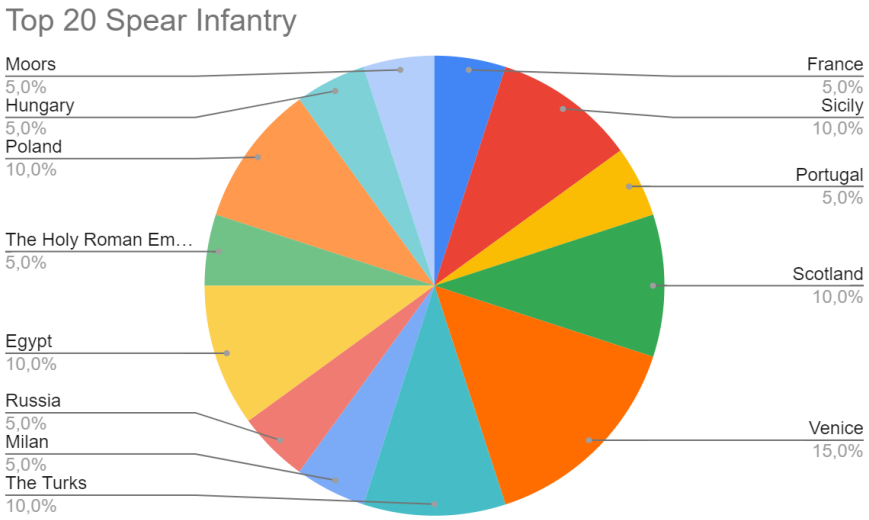

Figure 41. Distribution of Top 20 Spear Infantry Units to Factions.

Chapter 5

Discussions and Conclusions

An overview of the results, conclusions made from the study, and the suggestions for further studies on the field are presented in this chapter.

5.1 Summary of Findings

The first objective is to analyze efficiency between units of different factions. Findings reveal that there are many common units available for many factions such as Peasants, Town Militia, Spear Militia, etc. However, there are a few differences among those common units and the most visible one is about the Peasants units. I claim that Peasant units have been considered as the starting unit for each faction by the developers. According to playtest sessions, Peasants of The Byzantine Empire is the most damaging and the least damaged unit among the same units of other factions.

Table 8

Ranking of Every Peasant Units for Each Faction

159	Byzantine Empire	Peasants
160	England	Peasants
166	France	Peasants
177	Hungary	Peasants
186	The Holy Roman Empire	Peasants
193	Russia	Peasants
198	Milan	Peasants
209	Venice	Peasants
216	Spain	Peasants
221	Denmark	Peasants
222	Portugal	Peasants
230	Poland	Peasants
243	The Turks	Peasants
248	Sicily	Peasants
251	Moors	Peasants
264	Egypt	Peasants

Historically, The Byzantine Empire is the playable faction who was destroyed earliest among other factions. Even though all stats of Peasant units of Christian factions are identical, The Byzantine Empire has the strongest Peasants unit in the game according to playtest sessions. Another observation is that the Peasants of Islamic factions are further behind among Christian factions in the ranking. While Peasants of Christian factions cost 110 gold per unit, that number is 90 for Islamic factions and while Christian Peasants have 4 Attack value, Islamic Peasants have 1 Attack value. This change in costs and attack values between Christian and Islamic factions for Peasants is visible on the ranking also.

Battlefield efficiency of all those common units are noted after playtest sessions. Even though they have the same unit stats and playtest results were not exactly the same in all 3 playtest sessions of all common units against controlled unit Peasants, remaining soldiers after sessions were similar and consistent. The reason for having different remaining soldiers after every playtest session is other effects that could not be measured quantitatively during playtests which are mentioned in chapter 3. Concretely, the battle mechanics of Medieval II Total War is not about numbers and even the cheapest unit of the game which is Peasants are allowed to win a battle against more powerful opponents in the battlefield.

The second objective is to understand whether game unit balance is fair for all factions. According to findings in chapter 4, some factions have advantages in terms of units among others in some unit categories. While %20 of units in Top 20 Infantry belong to Spain, only %5 of units in Top 20 Cavalry belong to Spain. On the other hand, while France has no units in Top 20 Infantry, %20 of units in Top 20 Cavalry belong to France. To wrap up, ultimately unit inventories of factions have been designed to serve those factions in the campaign mode originally. Therefore, the concerns of campaign mode such as starting positions, needs or building units of factions have been ignored to focus on efficiencies of units in the battlefield. Nevertheless, dramatic advantages specific to unit diversity or effectiveness have not been observed in the playtest sessions.

The third objective is to analyze the power property differential between unit categories. Each unit type has advantages and disadvantages according to the opponent units category. The logic under that fact is very similar to the game Rock Paper Scissors. Rock crushes scissors, scissors cut paper and paper covers rock. For our scenario, spears are more valuable against cavalry, cavalry is more valuable against infantry, and infantry is more valuable against spears. According to findings in chapter 4, Top 20 units among all are cavalry units and that makes sense if we consider the controlled unit Peasants was an infantry unit. However, there are two more unit categories that are harder to place somewhere which are missile infantry and siege engine units. Missile infantry units have 2 weapon types. Their primary weapon is generally bow and javelin. Melee units should collide with an opponent unit to give it a damage but missile infantry units don't need a collision. They can give damage to the opponent unit with their ranged attacks until they run out of ammunition and while they are doing that, they also keep a certain distance between the opponent unit. That makes missile infantry units in a very advantageous position among infantry units. However, their ammunition is limited. Besides being very weak against cavalry units because cavalry units are much faster in the battlefield, they will have to switch to their secondary weapon which is generally short blades that cause low damage values and according to playtest results missile infantry units are not useful in 1 v 1 fight against infantry units. That places missile infantry units in a highly tactical position in battles.

Siege engine units have a very unique place among other unit categories in the battlefield. They are mainly designed to break city walls and their ammunition is limited and have secondary weapons just as missile infantry units. However, ranged attacks of siege engine units have a very huge impact on opponent units morale and ranged attacks of these units can cause multiple casualties at once. Although siege engine units are very slow and have a very low fire rate, they are as tactical as missile infantry units.

The fourth objective is to establish a combat prediction tool to predict combat outcomes. The tool we established in Python to predict combat outcomes gives conformable results with the results of playtest sessions for the infantry vs infantry unit matchings. Even though the equation gives largely compatible results with the

playtest results, it has only been tested on 1 unit vs 1 unit matching scenario. 2 units vs 1 unit or other combinations have not been tested in this study. However, by following the same method with predicting battle outcome of infantry units, it can be applied among other unit categories als well by adjusting equation variables according to playtest notes.

Final objective is to analyze whether faction strength and weaknesses are fair in the context of game units. Although the playtest results give an idea about advantages about factions, those inferences would be weak without considering other variables such as campaign mode scenario, starting positions of factions and even historical approaches to events of those factions. However, according to playtest results, some factions have superiorities in terms of unit rankings particular to some unit categories.

Spain is taking particular attention to the infantry category. %15 of the Top 20 Light Infantry units belong to Spain and Hungary is in a good position for the Light Infantry category.

France is very rich in Cavalry units. %20 of the Top 20 Cavalry units belong to France.

While Missile Infantry units of Muslim factions have better rankings than Christian factions, Muslim factions are weak in terms of unit efficiency on Cavalry and Infantry categories.

5.2 Conclusions

In this thesis, I have worked to evaluate unit balance and predict combat outcomes of two units' competition in the battle engine of Medieval II Total War. I mainly focused on unit efficiencies in the battlefield by considering and understanding unit stats to see whether all factions have similar starting positions among each other in terms of game units and fairness. My approach has been based on the data collected from playtest sessions, game community on the internet and the observations of units attritions in the battle engine of the game.

Processed data of playtest results has been used to give all units of factions a ranking based on their efficiency in the battlefield. However, this approach has some limitations. Main deficiency is that all the units have been fought against one only unit and the kill and death rate of a unit is strongly dependent on the opponent unit's infantry category besides many other variables mentioned in the thesis. For example, a Spear Militia unit might not be efficient against Peasants which costs less then Spear Militia but at the other hand, it might be more efficient against cavalry units. Nevertheless, the data collected from the playtest sessions will be a guide to analyze unit weaknesses against different infantry categories quantitatively by processing the combat outcomes of the units they have fought against Peasants.

Lanchester's Attrition Laws have been used to establish a combat prediction tool for the game in this thesis. Lanchester equations have been used to predict combat outcomes of RTS games such as StarCraft II and Age of Empire II before in some research. Main similarity of those two games is that the units have one individual soldier and also have much less unit stats than the Medieval II Total War to apply on the prediction tools. The prediction tool established in this thesis gives very similar results to playtest results even though accepting units as including multiple soldiers.

Comparison of the playtest results and the prediction tool outcomes has not been tested on the 2 v 1 battle scenario. The prediction tool can still predict the winner of the battle and can even predict the remaining soldiers of the winner and loser sides in the 2 v 1 scenario. However, the tool doesn't have the ability of predicting the remaining soldiers of 3 separate units individually. Lanchester's Square Law suppose that two armies shooting continuous bullet streams to each other. Each soldier in the army can aim at multiple targets and can receive multiple damage from multiple directions. Even so, the prediction tool established in this thesis supports that equation; it's not convenient to predict soldiers' aiming choice when there are two or more units in the opponent army.

Another limitation of the combat prediction tool established in this thesis is that the prediction tool still needs to be improved to predict the winner with high accuracy as it predicts melee vs. melee unit matchings in certain conditions such as the scenario

of Missile Infantry unit vs. Cavalry unit. As we know from the playtest notes, Peasants lost 65%-70% of their men at the collision moment with the cavalry units and on the other hand Peasants lost 20%-25% of their men before the collision with the missile infantry units. Our tool predicts the combat outcome by the hits units take. However, we don't quantitatively know the damage of the collision that cavalry units give to infantry units even if we can calculate the ranged attack damage of the missile infantry units.

According to findings in chapter 4, some factions are more advantageous in terms of unit inventory. However, I claim that it doesn't affect the fairness in the campaign mode as it affects the results on the battlefield. As mentioned before, ultimately unit inventories of factions have been designed to serve those factions in the campaign mode. Medieval II Total War is highly history dependent and many historical events occur at certain dates. Enemies or hostile states of the factions and the diplomacy at the time the campaign mode takes place historically consistent. Consequently, difficulty of a faction comes from historical events that the faction have faced rather than game balance factors in the campaign mode. Therefore, in my opinion advantages of some factions in the context of unit efficiencies and inventories should not be addressed to unfairness.

Another conclusion is that no strong and direct findings were found addressed to unfairness among factions even though some factions have more efficient units than the others in the Medieval II Total War game. I consider those advantages as components to stabilize campaign mode playability of some factions.

To summarize, prediction of the combat outcomes with the tool established in this thesis is convenient to the behavior of most game units in the battlefield. Doing playtest sessions was a good start to optimize the parameters of the prediction tool. Even though it has some limitations in certain conditions and needs to be improved, it will be a good starting point for the improvements that will cover wider conditions.

Consequently, this thesis introduces a method for evaluating complex game unit behaviors in games in which players control large scale armies such as Medieval II

Total War to improve the effectiveness of the unit balancing process for game candidates when there is no replay data of previous gameplays.

5.3 Recommendations for Future Research

Methods used in this study can be improved by integrating them into automated systems mentioned in chapter 2. Playtest sessions are time consuming when there are numerous game units to evaluate. Automating this process using search algorithms such as Monte Carlo Tree Search can accelerate this process.

Modding community of Medieval II Total War has wealthy solutions to improve the conditions of playtest environments and would help researchers to ignore variables hard to quantitate such as morale and the death moment of the unit general while making playtest sessions and even logic of some unit stats penalties such as shield and armor can be integrated into combat prediction tool established in this thesis for Medieval II Total War to acquire much more sensitive results. In this way, prediction results can provide a more inclusive situation for researchers to compare playtest results with prediction outcomes instead of comparing prediction outcomes with playtest results.

REFERENCES

Chen, H., Mori, Y. & Matsuba. I. (2014). Solving the balance problem of massively multiplayer online role-playing games using coevolutionary programming. *Applied Soft Computing 18*, 1–11.

Churchill, D., Saffidine, A., & Buro, M. (2012, October). Fast heuristic search for RTS game combat scenarios. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Elo, A. E. (1978). The Rating of Chessplayers, Past and Present, Volume 3.

Han, S. W., Lee, J. J., & Park, J. W. (2008). The Case Study on Game Balancing for Player VS Player Fighting Game. *Journal of Korea Game Society*, *8*(1), 15-27.

Jaffe, A. (2013). *Understanding Game Balance with Quantitative Methods.* (Doctoral dissertation). https://digital.lib.washington.edu/researchworks/handle/1773/22797

Karavolos, D., Liapis, A., & Yannakakis, G. (2017). Learning the patterns of balance in a multi-player shooter game. In Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG '17). Association for Computing Machinery, New York, NY, USA, Article 70, 1–10. https://doi.org/10.1145/3102071.3110568

Lanchester, F. W. (1916). *Aircraft in warfare: The dawn of the fourth arm*. Constable limited.

Makin, O., & Bangay, S. (2017, January). Orthogonal analysis of StarCraft II for game balance. In *Proceedings of the Australasian Computer Science Week Multiconference* (pp. 1-4).

Medieval II Total War. (2022, April 15). In Wikipedia. https://tr.wikipedia.org/wiki/Medieval_II:_Total_War

Stanescu, M. A., Barriga, N., & Buro, M. (2015, September). Using Lanchester attrition laws for combat prediction in StarCraft. In *Eleventh artificial intelligence and interactive digital entertainment conference*.

Stanescu, M., Barriga, N. A., & Buro, M. (2014, September). Hierarchical adversarial search applied to real-time strategy games. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Stanescu, M., Barriga, N. A., & Buro, M. (2017). Combat outcome prediction for real-time strategy games. In *Game AI Pro 3* (pp. 301-308). AK Peters/CRC Press.

Taylor, J. G. (1983). Lanchester Models of Warfare, Volume 1 and Volume 2.

The Complete EDU Guide. (2022, May 11). In twcenter http://www.twcenter.net/forums/showthread.php?111344-The-Complete-EDU-Guide

Uriarte, A., & Ontanón, S. (2017). Combat models for RTS games. *IEEE Transactions on Games*, *10*(1), 29-41.