Creating a Dynamic Battle System for a Massive Multiplayer Online Real-Time Strategy Game

João Luís Parracho Fazenda dos Reis

Thesis to obtain the Master of Science Degree in Information Systems and Computer Engineering

Supervisor: Prof. Pedro Alexandre Simões dos Santos

Examination Committee

Chairperson: Prof. Mário Jorge Costa Gaspar da Silva
Supervisor: Prof. Pedro Alexandre Simões dos Santos
Member of the Committee: Prof. Rui Filipe Fernandes Prada

October 2017
I would like to thank everyone that supported me throughout these years and that have helped me when I needed them, specially my family, without whom these project would not have been possible.

I would also like to acknowledge my dissertation supervisors Prof. Pedro Santos and Daniel Vila Boa for their insight, support and sharing of knowledge that has made this thesis possible.

A special thanks to my girlfriend who always lifted my spirits when I needed it and who did not let me discourage from doing this work, as well as being a constant joy in my life.

Last but not least, to all my friends and colleagues, new and old, that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.
Abstract

World War Online is a massive multiplayer strategy browser game made by Chilltime. This work’s objective was to improve the previous battle system by reworking it completely both in gameplay and user interface, so that players would feel like they had more control over the outcome of each battle, reducing the random factor, and also to increase the number of players actively playing the game. We start by analysing the previous battle system in detail and then we analyse other games’ with similar concept but with more robust battle systems. Afterwards, we describe the usability issues found on the previous battle system through user testing. As for our solution, we start by describing in detail its implementation and how it came to be by detailing the prototype evolution and its testing with users. Finally, we present the results obtained from this work for this new battle system.

Keywords

Battle system, strategy game, user interface, user testing.
Resumo

World War Online é um jogo de estratégia multijogador jogado num navegador de internet, criado pela Chilltime. O objetivo deste trabalho foi melhorar o sistema de batalhas anterior, reformulando-o completamente tanto em jogabilidade como em interface de utilizador, para que os jogadores sentissem que têm mais controlo sobre o resultado de cada batalha, reduzindo o fator de aleatoriedade, e para aumentar o número de jogadores que jogam ativamente o jogo. Começamos por analisar em detalhe o sistema de batalhas anterior e depois analisamos jogos com conceitos semelhantes mas que tenham sistemas de batalha mais robustos. De seguida, descrevemos os problemas de usabilidade encontrados no sistema de batalhas anterior, obtidos através de testes com utilizadores. Quanto à nossa solução, começamos por descrevê-la em detalhe e como é que ela foi criada, apresentando a evolução de protótipos e os testes feitos com utilizadores. Finalmente, apresentamos os resultados obtidos através deste trabalho para este novo sistema de batalhas.

Palavras Chave

Sistema de batalhas, jogo de estratégia, interface de utilizador, testes com utilizadores.
# Contents

1 Introduction .......................................................... 1
   1.1 Motivation .......................................................... 3
   1.2 Objectives .......................................................... 3
   1.3 Thesis Outline ...................................................... 4

2 Related Work ......................................................... 5
   2.1 Game Overview ...................................................... 7
   2.2 Map Interaction ..................................................... 8
   2.3 Attacking and Defending .......................................... 10
   2.4 Global Interactions ................................................ 11
   2.5 State of the art in similar games ................................ 12
       2.5.1 Map Navigation ................................................ 13
           2.5.1.A Browser .................................................. 14
           2.5.1.B Other platforms ........................................ 14
       2.5.2 Attacking and defending .................................... 14
           2.5.2.A Browser .................................................. 14
           2.5.2.B Other platforms ........................................ 15
       2.5.3 Defensive and offensive Support ............................ 15
       2.5.4 Battle System .................................................. 16
           2.5.4.A Browser .................................................. 16
           2.5.4.B Other platforms ........................................ 18
   2.6 Usability in Games ................................................ 18
       2.6.1 User Interface ................................................ 19
       2.6.2 User testing ................................................... 20
       2.6.3 Game analytics ................................................ 21

3 Previous Battle System .............................................. 23
   3.1 Detailed overview ................................................ 25
   3.2 Usability testing ................................................... 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1 Inconsistency</td>
<td>30</td>
</tr>
<tr>
<td>3.2.2 Cryptic</td>
<td>31</td>
</tr>
<tr>
<td>3.2.3 Inefficiency</td>
<td>32</td>
</tr>
<tr>
<td>3.2.4 Error inducing</td>
<td>32</td>
</tr>
<tr>
<td>3.2.5 Recognition rather than recall</td>
<td>32</td>
</tr>
<tr>
<td>3.2.6 Not providing feedback</td>
<td>32</td>
</tr>
<tr>
<td>3.3 Strengths and Weaknesses</td>
<td>33</td>
</tr>
<tr>
<td>3.4 Conclusions</td>
<td>33</td>
</tr>
<tr>
<td>4 Proposed Solution</td>
<td>35</td>
</tr>
<tr>
<td>4.1 Final Design</td>
<td>37</td>
</tr>
<tr>
<td>4.2 Solution Architecture</td>
<td>42</td>
</tr>
<tr>
<td>4.2.1 Server Side</td>
<td>43</td>
</tr>
<tr>
<td>4.2.2 Client Side</td>
<td>45</td>
</tr>
<tr>
<td>4.3 Implementation and Testing</td>
<td>45</td>
</tr>
<tr>
<td>4.3.1 Version 1</td>
<td>46</td>
</tr>
<tr>
<td>4.3.1.A Usability analysis</td>
<td>48</td>
</tr>
<tr>
<td>4.3.2 Version 2</td>
<td>49</td>
</tr>
<tr>
<td>4.3.2.A Usability analysis</td>
<td>50</td>
</tr>
<tr>
<td>4.3.3 Final Version</td>
<td>50</td>
</tr>
<tr>
<td>4.3.3.A Usability Analysis</td>
<td>52</td>
</tr>
<tr>
<td>4.3.4 Conclusions</td>
<td>52</td>
</tr>
<tr>
<td>5 Final Results and Conclusion</td>
<td>53</td>
</tr>
<tr>
<td>5.1 Data discussion</td>
<td>55</td>
</tr>
<tr>
<td>5.2 Conclusion</td>
<td>57</td>
</tr>
<tr>
<td>5.3 Future Work</td>
<td>57</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Map View of Europe</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Window Open Example</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Map Zoomed View</td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td>Map District View</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>Available Sections</td>
<td>12</td>
</tr>
<tr>
<td>2.6</td>
<td>Map view of games similar to WWO played on a browser</td>
<td>13</td>
</tr>
<tr>
<td>2.7</td>
<td>Map view of games similar to WWO played on other platforms</td>
<td>15</td>
</tr>
<tr>
<td>2.8</td>
<td>Battlefield in Forge of Empires and Command &amp; Conquer: Tiberium Alliances</td>
<td>17</td>
</tr>
<tr>
<td>2.9</td>
<td>Battlefield view of games relevant for World War Online (WWO)'s new battle system</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Unit battle stats</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Battle first step</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td>Battle second step</td>
<td>27</td>
</tr>
<tr>
<td>3.4</td>
<td>Battle report</td>
<td>29</td>
</tr>
<tr>
<td>3.5</td>
<td>Usability Issues</td>
<td>30</td>
</tr>
<tr>
<td>3.6</td>
<td>Nested window</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Attack flow</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Attack first step</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>Attack position units step</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>Battle report list</td>
<td>41</td>
</tr>
<tr>
<td>4.5</td>
<td>Battle replay</td>
<td>42</td>
</tr>
<tr>
<td>4.6</td>
<td>Battle flow</td>
<td>43</td>
</tr>
<tr>
<td>5.1</td>
<td>Battles fought</td>
<td>55</td>
</tr>
<tr>
<td>5.2</td>
<td>Damage Dealt per battle</td>
<td>56</td>
</tr>
<tr>
<td>5.3</td>
<td>Units killed per battle</td>
<td>56</td>
</tr>
</tbody>
</table>
List of Algorithms

3.1 Battle Processing Main Flow ............................................. 28
Acronyms

**MVC**  Model-view-controller
**UI**    User Interface
**RTS**   Real-Time Strategy
**MMORTS** Massive Multiplayer Online Real Time Strategy
**WWO**   World War Online
**AI**    Artificial Intelligence
1

Introduction

Contents

1.1 Motivation ................................................. 3
1.2 Objectives .................................................. 3
1.3 Thesis Outline ............................................. 4
This thesis was integrated in a company, Chilltime, whose main product is World War Online (WWO). WWO is a Massive Multiplayer Online Real Time Strategy (MMORTS), currently browser based, built using mainly PHP and Javascript. It’s similar to several established games, such as the Total War series [1], GoodGame Empire [2] and Forge of Empires [3], in terms of overall concept, game mechanics and/or interface.

The game is divided in 3-4 seasons per year and, in the beginning of each season, every player has to build up from the start. Players can play alone or they can get together in squads to play collectively, specially because only by joining a squad can you participate in the main objective and enjoy the full experience that the game provides. The game’s main objective is to capture and hold countries’ capital locations to earn points for your squad, and be the squad with the most points by the end of each season. At the end of the year, the country which, on average, got the highest positioned squads, is deemed the winner. The players can, and are encouraged to, join a squad from their real country but can also join squads from other countries if they wish to.

WWO appeals to players mainly through two key aspects, the community interactions, which consists mainly on participating on the forums, sharing battles and discussing strategies, and its real-time battles between players.

This thesis will focus mostly on the battle system, it’s strengths and weaknesses and how we propose to improve it.

1.1 Motivation

With this work, we want to create an experience for players where they feel like they have meaningful choices to make, and that by making them they can overcome even situations where the enemy has the upper hand. Currently, you would either win by a landslide or rely on luck. So, we decided that a new battle engine should be created, one that would allow players to strategically position their forces and be able to customize their behaviour, changing the way they would choose their targets and allowing them to have unexpected moving or attacking patterns.

1.2 Objectives

The main goals we want to achieve with this work are:

1. increase the sense of control on each battle;

2. create a battle system that allows user input in different stages of the same battle.
By accomplishing these main goals, we also want to increase the average number of battles made by players and increase the number of players actively playing the game. For these two secondary goals, great increase values would be 10% and 25% respectively.

In order to check if these goals are achieved, we will be doing user testing, both in-house and on public events, and analyse Chilltime's detailed set of metrics.

1.3 Thesis Outline

We will start our thesis with a related work chapter where we will give an overall description of our game, what a player can and can’t do, followed by a research of similar games, whether they are browser based, desktop based or mobile based. This research will focus on how their interfaces are built and how successful they are in achieving a particular goal, how their battles work, which options are supplied to players and how players view and/or interact with the battle itself. Then, we will introduce some concepts about User Interface, user testing and usability that are required to better understand our evaluation process.

Secondly, we will do a thorough analysis of the previous battle system, starting by giving a detailed overview about it, describing the elements it is made of, like units, algorithms and interface. Then we will present the usability issues found in that interface, sorted by category. Lastly, we will conclude what are this battle system's strengths and weaknesses.

Afterwards, we will describe the proposed solution, detailing which new elements were introduced to the battle system, both in behaviour and interface. Then we will detail how the system is built both in server and client sides, what is done on each side and the structure that was used. We then describe the prototype evolution, describing what each prototype was and the problems that we found through user testing and how they were fixed in the following prototype version, as well as how the user testing was conducted.

Finally, we discuss the results obtained and how they met, or did not meet, our goals and why.
Related Work

Contents

2.1 Game Overview ..................................................... 7
2.2 Map Interaction ..................................................... 8
2.3 Attacking and Defending .......................................... 10
2.4 Global Interactions ................................................ 11
2.5 State of the art in similar games ................................. 12
2.6 Usability in Games .................................................. 18
In this chapter, we will give a detailed overview of the whole game to better contextualize our decision process, and we will be analysing WWO's key features and comparing them with similar games that fit the same genre, in order to better understand the state of the art of games inside the same genre and compare their solutions with ours. We will also explain some concepts required to understand our evaluation process.

### 2.1 Game Overview

We will start by giving a global overview of the game, what is the objective, how the player interacts with the world that is presented to him, which actions he is able to take and the core game flow. We will then go more in depth on topics related with the battle system, which is the main focus of this work.

The game world is not persistent and is reset every 3-4 months. This time period between resets is called a season and there is a leaderboard for each season. When the last season of the year is finished, all countries’ scores for each season are taken into account and a final podium is created, announcing who are that year’s winners. When each season starts, each player has to build their bases and armies from scratch to ensure that everyone starts with the same relative strength, although players remain on the squads that they were on in the previous season. The main objective throughout the season is to win the championship by scoring points for your squad and country by conquering available countries’ capital locations and be the squad that scored more points. Therefore, joining a squad is one of the most important steps of the game, as you are required to be on a squad to participate in the championship, whether that squad is fighting for your country or any other country.

As for gameplay, the player can move around a world map, on which he can build resource bases to produce resources and build his army. Then, he uses his army to vanquish his opponents and support his allies to capture the countries’ capitals, to earn points and win the championship. He can also attack his opponents’ resource bases to steal resources, weakening his foes and boosting his army production. The battles globally consist on selecting an enemy target base or army and analysing the enemy’s defence, followed by selecting some of your units to attack with a particular strategy.

The main view of the game is the Map View (Figure 2.1), which is where the player can execute the most relevant actions and where the player can access most of the actions available with the fewest clicks. With exception of actions executed on the map itself, when a player executes an action, it will open a window which will contain all the relevant information for him, and any further possible actions (Figure 2.2).

As a player needs to prepare his army and strategy to be able to win, players usually spend their time either preparing for war or fighting one. When preparing for war, players spend most of their time creating diplomatic relationships, producing and stealing resources and then building their army for an upcoming
As a consequence of diplomacy, squads can ally with each other, with each squad only being allowed to form an alliance with other three and, by being allies, these squads’ players can coordinate themselves to capture objectives and to fight each other’s enemies, effectively increasing their chances to successfully capture objectives and climb the leaderboard.

On the other hand, when fighting wars, players mostly focus on which armies to attack, which bases to conquer and how to overcome the enemy defences. Players also need to rebuild their army and move armies between locations, either to attack in multiple fronts or to safeguard some of their army and only use it when the best time comes.

### 2.2 Map Interaction

The map is where the game is played and here a player can take many actions, such as changing continent, moving units, selecting districts, attacking players, conquering capital bases and check the game current status as for who currently owns each capital.

The map starts out by being zoomed out, enabling the player to see most of the continent he is currently on (Figure 2.1). There is another level of zoom that (Figure 2.3), although it doesn’t provide any additional information, allows an easier selection of existing elements. The player can then click on his army flags or on any district to obtain more detailed information about that army or district, respectively.

When selecting a district (Figure 2.4), the player will now be able to see all the bases that have been
Figure 2.2: Example of an opened window (Community - Capital Warfare section).

Figure 2.3: Map zoomed view. Although not providing any additional information compared to the view presented in Figure 2.1, it allows a better understanding of the map and easier clicking on its elements.

built there, the relative military strength of the player compared to the base’s owner and to which squad does the owner belong to, which are shown by specific icons, color code and by displaying the squad’s
photo, respectively.

On the small window to the right, he can also see the army he has in that district, if any, and check who else has an army there. By clicking on a base or army, the player can see who owns it, his military rank, his squad and his defending army. The player can never see the entire army that he is going to attack, only being able to see the portion of the army that he is going to fight if he chooses to attack. As the player attacks and destroys portions of the armies, the remaining defending army automatically replaces the units destroyed until there is no defending army left.

As for actions a player can take on a base, if it’s a squadmate’s base, he can support the defending army, if it’s an enemy base, he can attack, conquer or destroy it and, if it’s his own base, he can train units, choose units to defend the base and upgrade or destroy the base.

![Figure 2.4: Map district view. White and red bases are owned by enemy players with similar and higher military strength, respectively and green bases are owned by players from your squad. The small window on the right has more detailed information about the player’s and enemies’ armies and bases.](image)

2.3 Attacking and Defending

Attacking a base will earn the player experience points, steal resources, if available, and possibly earning points towards the leaderboard. *Conquer* and *Destroy* are similar actions to that of a basic attack but, on a successful *Conquer*, the attacker will become the owner of the base and on a successful *Destroy*, the base will cease to exist. Both conquering and destroying have a cost, whereas attacking
costs nothing. If the target is an army, a player can only attack it and he won’t steal any resources. This is usually used to weaken the enemy and prevent him from attacking you or your allies.

After analysing the enemy’s defences and possible rewards, the player can now decide if he wants to attack any base or army and, if he chooses to do so, he will have to plan his strategy so that he will win with the least losses, based on the enemy’s defence, which he has had the opportunity to analyse.

We will now give an overview of how the player makes an attack and what choices are presented to him, but this will be detailed further in Section 3.1. First, the player chooses up to 5 of his units to attack with (Figure 3.2). Secondly, he chooses their strategies, which will modify how they behave in the battlefield (Figure 3.3). There are four different strategies available which are described in detail on chapter 3.1. Finally, the player sends his attack which will take 3 minutes to hit the defending army. When the attack hits, all players involved in the battle receive a battle report which details what happened in the battle on each wave and the overall result, the damage done and the winner’s rewards and the loser’s losses (Figure 3.4).

When defending, a player can choose 5 different units in the quantity that he wishes, as long as he doesn’t exceed his military budget, which increases with military strength and depends on whether the units are Supreme Units or Normal Units. A player's base or army can then be supported by an ally with another 5 different units with a separate military budget.

### 2.4 Global Interactions

Besides the map actions, there are several other actions that are available at any time and that the player uses to learn more about the players surrounding him and the game itself. The player can communicate with others through the chat and message box (Figure 2.5 - 1 and 2), enabling him to ask for help or taunt other players, as well as asking for advice from more veteran players or discussing diplomacy with leaders from other squads.

He can also check his current resources, resource production and army upkeep (Figure 2.5 - 3), which is essential to manage your economy and to know if you have enough resources to sustain your armies, otherwise they will start dying. As for knowing your surroundings, he can access other player’s profiles to check which country they are from, how strong they are and if they belong to a stronger squad as well as to check their performance on previous seasons. Furthermore, he can check countries’ profiles, check his own profile to see how is he doing, check squads’ profiles to see if they are strong or not, who its members are and apply to join them if they have available slots (Figure 2.5 - 5).

To manage how he is doing and to improve his state, the player has, on the bottom, the main action buttons that allow him to: access all information about his bases, as well as upgrade them; know his army size, composition and location, so he knows what he is missing and where he can fight; access his
Figure 2.5: Sections that are always available to the player. 1) Chat; 2) Message box; 3) Current resource and resource production; 4) Base, army, battles, and leaderboard buttons; 5) Player Profile and information; 6) Settings, help, guides and community.

battle history to know on which battles he participated and the result of each battle, to keep up-to-date on the success rate of the his attacks and defences as well as know what he has gained and lost with each battle, including leaderboard points; and access the leaderboard where he can actually see his own standing or his squad's and compare it with other players or squads (Figure 2.5 - 4).

Finally, on the top right corner of the screen, there are other actions that are not essential for playing the game but some of which are a core part of the game as players use them often to interact with each other as a community. These are: account settings, where the player can change his name, avatar and account information; help, where a player can report bugs or other players or submit a feedback report to help the developers improve their game; guides, where all rules and information about the game are gathered and is where players go to when they have doubts about how something works; and community sections, which include epic battles submitted and voted on by players, and forums, where players can talk to each other, leave tips or just taunt each other (Figure 2.5 - 6).

2.5 State of the art in similar games

In this section we will be comparing some of WWO's key features, such as map navigation, attacking and defending and the battle system itself, with other strategy games that have a similar audience to
2.5.1 Map Navigation

Although the map in WWO is not directly related to the battle system itself, it’s a key element for a player to find a target and to make an attack, making it vital for the battle system to work. It is where players spend most of their time and where they execute most of their actions, as it is where they can defend their troops, search for enemies to attack and keep track of players of interest.
The map is always present, whether it is the main focus or if it is partly occluded by an overlapping window. The player can navigate the map by dragging it with the use of his mouse and can interact with most of its components by clicking on them.

2.5.1. A Browser

Many of the most played browser strategy games, like Tribal Wars 2 [4], GoodGame Empire [2] and Command & Conquer Tiberium Alliances [5], have a very similar map view, where you see villages, castles and bases, respectively (Figure 2.6 - a, b and c), instead of a grid of hexagons, but the player also drags the map to navigate it and clicks on target structures to send an attack or to investigate its defences.

Nowadays, this is a fairly common way to navigate a map and display information to the player as it provides a vast range of targets that a player can choose from, since the map can be larger than the user’s screen. Also, by using distinct shapes and colours, these targets can easily be distinguished from each other.

2.5.1.B Other platforms

On other platforms there are also many strategy games on which the player navigates through huge maps and manages armies, resources and bases. Some examples are Endless Legend [6], Total War: Warhammer [1] and Planar Conquest [7] (Figure 2.7). In all three, the player has access to a map where he can see his and the opponent’s armies and bases as well as other points of interest. To select armies and take actions the player uses the mouse and clicks on what he wants to target, much like WWO. However, to move around the map the player needs to bring the mouse to the edges of the screen instead of clicking and dragging the map itself.

2.5.2 Attacking and defending

These are the most important features of WWO as they allow the player to have the most impact. Attacking and defending successfully will make the country the player is fighting for reach a higher position in the leaderboard by weakening the player’s enemies and by obtaining points for his country.

2.5.2.A Browser

Most strategy games similar to WWO which have a battle component have a similar approach to it - building units, selecting units to battle, giving selected units some strategies to follow or items to use, and then the battle is processed automatically.
On one hand, similarly to how WWO is played, in GoodGame Empire, to attack, you build your units and then select a target on the map. Then, you select which units you wish to attack with and which support item should they use to improve their chances of winning. OGame [8] has also a similar approach, but you only choose the units to attack, having no extra input to increase your chances of winning other that selecting the right units. In both of these games, in the end, you receive a static battle report, detailing what happened on the battle.

On the other hand, with different and more interactive approaches we have Forge of Empires [3], on which you also select the units to counter the enemy’s defence but then you have a dynamic turn-based battle, where you can opt to command your units to do what you want or you can auto-resolve the battle. As you have a dynamic battlefield, you receive no battle report in the end. This dynamic battlefield is close to what we want to achieve with this work.

2.5.2. B Other platforms

On other platforms, in games like Planar Conquest, Total War: Warhammer and Endless Legend, to attack you do something similar to what you do in WWO, which is bringing your army to the same position where the opponent’s army is and then you prepare to attack. Unlike what WWO does right now but more towards our goal, these games have a dynamic battlefield where both the player and his opponent take turns to control each own units synchronously, instead of fighting an Artificial Intelligence (AI).

2.5.3 Defensive and offensive Support

Supporting your teammates, whether offensively or defensively, is also key in achieving victory in WWO. Only by doing so can you overcome someone’s defences or strongly secure an important objective. It also strengthens the bonds between players as they feel gratified by helping others and by being helped. In WWO, to support offensively an attack has to be already on its way for you to be able do add your own troops to it, whereas defensively you can support any of your teammates whenever you wish.
On other browser strategy games, the presence of these features vary, mostly only being available a defensive support option. OGame has both these features and they work identically to WWO while GoodGame Empire only has a defensive support option, letting you send troops to help defend friendly castles.

This is a feature that isn’t very common on other platforms and games mentioned for those platforms, as these games use a more synchronous gameplay where all players require to be on-line at the same time and therefore each player, when wanting to help another, just moves their armies to protect another player’s bases or destroy other armies attacking such player.

2.5.4 Battle System

This is the feature that should allow players to show off their skills in battle, although in WWO battles were mostly won by sheer number of units and their strength versus one another. Based on the player opponent’s defence, the player has to carefully pick his units so that most of his units are the enemy’s weakness since the battle will then make them fight each other randomly. The battle system picks each of the attacking units to attack a defensive unit picked at random and this defensive unit retaliates simultaneously, which means, to win the player needs his units to take less damage from the enemy’s units and deal more damage to them. The strategies the player is able to choose for each unit influence this outcome slightly.

As this is the feature that we aim to improve with this work, we will present battle systems on other games which provide dynamic battles and that allow players to have meaningful inputs on units’ behaviours. Although some of the battle systems that are going to be analysed are much more complex than what we want to accomplish, they all have some components that can be adapted into our new battle system.

2.5.4.A Browser

Most browser games do not have very complex battle systems due to their limitations in processing power and their target audience. However, there are some games that present the player with dynamic battlefields which take into account the player’s input in order to change the outcome of the battle, allowing skilled players to perform better.

Forge of Empires is one of these games where the player enters a dynamic battlefield and can choose to give his input in order to perform better (Figure 2.8(a)). The attacking player controls all his units until the battle is finished and the defending player’s units are controlled by an AI. Each side takes their turn, by making units move or attack another if in attack range. The battlefield also has environment that influences these choices depending on the unit, as there are units that move better than others or get an attacking or defensive bonus by being in certain environments.
(a) Forge of Empires battle - attacking units on the left side (green) and defending units on the right (red)

(b) Command & Conquer: Tiberium Alliances - bottom section is for setting up offence and the upper section displays the defence.

**Figure 2.8:** Battlefield in Forge of Empires and Command & Conquer: Tiberium Alliances.
Similarly, we have Command & Conquer: Tiberium Alliances, which allows the player to customize his initial strategy but once the battle has started the player cannot give any more input and the battle will be processed until the end (Figure 2.8(b)). The battlefield has nine lanes on which the player can send his troops and the defender has positioned defences. The player analyses these defences and selects which units he wants on each lane and by which order they should enter the battlefield, being able to choose whether they should enter immediately or after 10, 20 or 30 seconds after the battle has started.

2.5.4.B Other platforms

We found many games that used different battle systems through different genres each one adapted to the games audience and theme, both single-player and multi-player. From those, we will detail a few features that we thought could be pertinent to our new battle system.

XCOM 2 [9] has very interesting behaviours that the player can select for his characters. It is a turn-based game and on your own turn you can move your units, attack or take a special action. These special actions are what is particularly relevant to us. First, an action which causes the unit to skip attacking this turn but instead if an enemy comes into range on the enemy’s turn, this unit will shoot it. Second, an action which causes the unit to skip attacking as well but that will, in exchange, grant an additional bonus to defence to that unit. Finally, there are also actions specific for a set of weapons which allow to turn the tide and increase the tactical value of the game.

In Total War: Warhammer, units can be ordered to hold their position and not pursue disengaged targets, although still engaging in combat if they come close. They can also be ordered to attack an enemy while keeping their distance allowing them to hit the enemy without letting them close in on them.

Fire Emblem [10] and Advance Wars [11] both have a similar way to play which consists on moving your characters/units through a battlefield made of square tiles (Figure 2.9(a) and Figure 2.9(b)). Selecting your units shows their attack and move range which is extremely helpful when planning what you are going to do on your turn. This battlefield style is what would probably suit our game best since its simple and effective allowing players to quickly understand what to do. Fire Emblem provides an interesting mechanic where, if you have the character you are attacking with next to another character on your team, this second character will assist in the attack increasing the effectiveness of the attack.

2.6 Usability in Games

User Interface (UI) usability is critical for a game to succeed and being usable is far more important than looking nice. It is also important that it looks good in the end but if an UI is cluttered and not easy to use, the player will feel lost and won’t be able to enjoy the game, no matter how good it might
be \cite{12}. Therefore it is of the utmost importance to test with real potential users and not only with the development team as their objectiveness might be compromised for being too familiar with the interface and too close to the game to provide objective feedback. It is also ideal to start testing early in the development because the later you get in the development process, the more costly it is to fix errors and therefore the less likely it is to fix them \cite{13}.

2.6.1 User Interface

An UI is what allows players to take action on a video game. Without any UI a game wouldn’t be playable as the player would have no way to give orders to the machine that is running the game. Players use both physical and visual interfaces, with physical interfaces being, in this case, keyboard and mouse as the player is playing on a computer, and the visual interface is all the buttons and items that the player can click on the screen or that display some kind of relevant information to the player \cite{12}.

As stated in \cite{12}, an interface is considered dysfunctional when it is:

- Cryptic - containing unintuitive controls or obscure graphics due to aesthetic over-enthusiasm.
- Complex - providing too many options so that a starting point is unclear.
- Simplistic - limiting player choice by including few components.
- Inconsistent - confusing players with style and logic clashes between the interface and the game itself, or within different areas of the interface.
- Inefficient - forcing players to interact too many times with the interface before they get to their destination.
- Cluttered - taking up too much screen space and obstructing the game content area, or incorporating crowded controls that are physically difficult for some players to use.
To create a great game interface, it is also suggested to keep the following guidelines in mind [12] [14] [15]:

- Be consistent - When using the same action in different situations, assign it to the same control. Also names and colours of things that appear in indicators and menus should be identical everywhere it’s used.

- Enable shortcuts for hardcore players - After players get acquainted with the game, they require ways to accelerate accessing common actions and ways to re-assign the keys used in the process.

- Provide feedback - Every action a player takes needs to provide any feedback to the player, visual or audible. This is true even when the player attempts to execute an action that is marked as disabled/locked.

- Permit easy reversal of actions - Unless it would affect the game balance negatively, the player should be allowed to undo his actions even if for a limited number of times.

- Allow for player control - When the player is controlling a character, it should do exactly what the player requests him to do and not any unexpected behaviour.

- Recognition rather than recall - Keep it simple and don’t strain the player’s short-term memory. Allow players to look up the information he needs or display the information he constantly needs on a screen element.

- Include a context-sensitive pointer - When selecting an object of interest or a screen element, make the pointer change colour or format.

- Use established conventions - Do not innovate unnecessarily if there already is a good solution widely used in other games that fit your genre.

### 2.6.2 User testing

User testing consists on observing your users using your interface, whether on a controlled environment or not, with both methods wielding different results. On controlled environments, users are usually given specific tasks to complete and are observed while doing so, whereas on uncontrolled environments, users are expected to use the interface as they would normally do [16]. Also, users might feel pressured on a controlled environment and do some things that they otherwise wouldn’t, and in an uncontrolled environment it might be hard to take notes of everything the user is doing, as well as controlling distractions that might interfere with the user’s focus.

With that in mind, we conducted tests with users regarding the previous battle system interface in order to be able to detect which problems the interface had. As we developed the prototypes for the new
battle system, we also kept making user testing whenever we felt it justified, in order to keep iterating as soon as possible and waste as little time as possible developing what wouldn’t work.

Details of these analysis can be found in Chapter 3.2, for the previous battle system, and Chapter 4.3, for the prototypes of the new battle system.

2.6.3 Game analytics

To complement user testing, we kept track of several metrics regarding players’ actions such as which buttons were pressed, which units were used on each battle and which units had the most victory to defeat ratio, the number of battles done, ratio of victorious to unsuccessful battles, amount of damage done per battle, among others. These metrics allowed us to analyse further the user testing, even if the testing was done in our presence. For example, if most players are not using strategies, we can assume that they are not meaningful to them or that they are not understanding what they do. However, to modify a unit’s strategy, the player has to press a button to show him the options available and, by keeping track if the player pressed that button to show the strategies, we can check if the players are even pressing the button and accessing the strategies selection menu or if they are not using strategies because they are not even aware that there are strategies to select.

We can also use these kind of metrics to find out if any of the units available is too strong or too weak by analysing their victory to defeat ratio and their overall usage. If a unit is being used much more frequently than any other or if it has a victory to defeat ratio much higher that expected, we can then turn our attention to that unit in particular and analyse the battles on which it was used to check if that unit is indeed to strong, if there is any bug with that unit or if other units that would counter this specific unit are too weak and ineffective.

In conclusion, these metrics are used to back up our ideas and/or to tear them down, as well as turn our attention to potential problems the players might not be reporting or even to find the real root of a problem which players claim there is with a certain feature. Many times players don’t realize everything they do and these metrics help us keep aware of what they aren’t realizing.
3

Previous Battle System

Contents

3.1 Detailed overview .......................................................... 25
3.2 Usability testing ............................................................. 29
3.3 Strengths and Weaknesses ............................................... 33
3.4 Conclusions ................................................................. 33
In this chapter, we will be analysing the previous battle system. We will describe how it all works in detail and, on one hand, how it successfully played its part and allowed players to have a global battlefield where they could effectively battle each other and, on the other hand, how it failed to provide a fulfilling experience.

3.1 Detailed overview

The previous battle system is very simple, not allowing players’ skill to be a determining factor on winning or losing battles. Globally, it considers a set of units the attacking player has chosen and puts them against another set of units from the defending player, processes the battle and then sends a battle report to all players involved.

Units can be Normal or Supreme and each unit fits one of four different types, infantry, armoured, air or navy. Normal units can be built without any special currency and are intended to be build in large numbers whereas Supreme units are built with the game’s premium currency, have stronger statistics and are intended to be built in small numbers. All units have a set of statistics that are common between all of them but that have different values depending on the unit. These statistics are: Health, Damage and Defence against each type of unit and special defence versus Normal or Supreme units (Figure 3.1). There are other side statistics but these have no influence in the battle system. Units also have a budget cost based on the unit’s cost which is used to limit the amount of units you can select for a battle.

![Figure 3.1: Example of a unit’s battle stats, as it’s shown to a player. From top to bottom we have health and attack/defence versus infantry, armored, air, navy, normal and supreme units.](image)

When a player decides to attack another, he is taken to a unit selection screen, where he will decide which units he is going to take and how many of each, up to a maximum of five different units and maximum budgets for normal and supreme units (Figure 3.2). The player does this by using his mouse...
and optionally keyboard to make it faster. Each unit is displayed as a unit card with a number input box below it. By clicking on the card, the maximum number possible of that unit is added to the selected units’ pool, if no amount of that unit had already been chosen or, if it had, the total amount selected for that unit is removed. In addition to clicking on the card, the player can press either vertical arrow that are on the right side of the input box, up and down, to increase or decrease the amount selected for that unit, respectively, or he can just type a number on the input box corresponding to the amount of units he wishes to select. Each time a player changes the amount selected for any unit, a status section to the right is updated, to show how much of the allowed budget is filled and how many different units are selected out of the total allowed. It also shows the amount of resources that the units selected can steal, if applicable.

Figure 3.2: The first step in sending an attack. Here the player chooses which units he wants to take into battle and how many.

On the next step, all units are given a default strategy but the player can select a different strategy for each unit, if he chooses to do so (Figure 3.3). These strategies can influence the unit’s behaviour in battle and/or change their statistics slightly. The strategies available are the following:

- **Default** - Units with this strategy attack each wave of the battle and retreat if their numbers are reduced below 30% of the starting number.

- **Cautious** - same as the Default strategy but units will retreat if their numbers are reduced below 70% instead of the 30% of the Default strategy. Units will also have a decrease of 15% on damage dealt and an increase of 5% on defence.
- **To The End** - same as the *Default* strategy but units will not retreat. Units will also have a decrease of 2% on defence and an increase of 5% on damage dealt.

- **Flank** - Units with this strategy will only participate from wave 5 onwards, not being target-able until then and dealing no damage. On wave 5 they behave in the same way as they would with the *Default* strategy, with the exception that they won't retreat. They will also have a decrease of 10% on defence and an increase of 25% on damage dealt.

![Figure 3.3](image.png)

*Figure 3.3*: The second step in sending an attack. After selecting his units, the player can modify their behaviour by selecting a different strategy for that unit.

After the strategies step, the player can then send his attack or simulate the battle. If he chooses to simulate the battle, the player will get a simulation report immediately, which is very similar to a regular battle report and will allow him to have a glimpse of what the battle result can turn out to be. As the battle processing has some random factors, the player can do multiple simulations and get an idea of what his chances of winning are. On the other hand, if the player chooses to attack, the attack will now be pending and will be processed in three minutes. The units he selected will no longer be available for any action while the attack is pending and the defending player will be warned of an incoming attack. During this period the defending player can change his defence if he wishes or flee with his army but there is no way that he can know what the attacking player has sent.

While the attack is on its way, both attacker and defender can be supported by their squadmates, which consists on adding additional units to the attack or defending side, up to a maximum of 5, making battles as big as 10 units on each side.
When the attack timer reaches zero, it’s time for the server to process the battle (Algorithm 3.1). It starts off by getting the attacking and the defending unit groups and then it processes each wave, up to a maximum of 50. On each wave, every attacking unit is matched with a defending unit and deals damage to it. At the same time, that defending unit retaliates and deals damage to the corresponding unit. This means that, on each wave, the defending side attacks as many times as the number of attacking units whether there are more attacking than defending units or vice-versa. For example, if there are 2 attacking units and 5 defending units, each attacking unit will attack a random defending unit and only the defending units that are attacked will strike back through retaliate. If the same defending unit was attacked by multiple attacking units, this unit will retaliate multiple times in the same wave. This means that both sides will only attack as many times as there are attacking units.

Damage from a unit is calculated in the following way: Suppose we have a normal infantry unit A attacking a supreme armoured unit B, where unit A has an attack value of 100 versus armoured units and B has 50% and 15% defence versus infantry and normal units, respectively. Damage dealt from A to B is $100 \times (1 - B\text{'s defence (ranges from 0 to 1)})$. B’s defence is calculated as 60% of B’s defence versus infantry plus 40% of B’s defence versus normal units which is equivalent to 60% of 50% plus 40% of 15% which is equal to 25% plus 6% which leads to B’s defence being 31% versus A. Therefore, B would take 31% less damage of the 100 that A would deal, meaning it would take 69 damage ($100 \times (1 - 0.31)$). Each unit has also a critical hit chance versus a single type of unit which doubles their damage output.

**Algorithm 3.1: Battle Processing Main Flow**

```plaintext
begin
maxWaves ← 50
currentState ← 1
unitListDefend ← getUnitListDefend()
unitListAttack ← getUnitListAttack()
while currentState <= maxWaves do
  foreach unit ∈ unitListAttack do
    targetUnitDefend ← getRandomUnitDefend(unitListDefend)
    attackTargetUnit(targetUnitDefend)
    retaliateDefendUnit(targetUnitDefend, unit)
  prepareNextWave()
end
```

In the end, after the battle has been processed, all players involved receive a battle report detailing what happened in the battle. The battle report list view allows players to quickly scroll through their latest battles and check who won, what was lost or won in the battle including resources, leaderboard points and units. It also allows to see who were the attackers and defenders and in which district it took place. There are also further options on each battle report, allowing you to see a battle video of the battle itself, its details and to share it to the chat or to submit it to the epic battles section.
The battle video consists in showing all units that participated on the battlefield and the actions they took on each wave through sprite animations. The units are stationary and each action they take is shown through a visual cue like a missile fire animation when a unit is attacking or a retreat message over the unit to indicate that they are now retreating. This battle video also has additional options like selecting a specific wave and cycle between the information being shown, damage dealt/taken ratio, amount of unit lost or current units versus starting amount.

The details section works like a summary of the battle video. It shows damage dealt and taken by all participants, percentage of units survived and all rewards won on each side.

![Figure 3.4: The battle report of a finished battle. Here we can see the battlefield (as shown) or we could choose to see statistics of the battle.](image)

### 3.2 Usability testing

To acknowledge what was wrong with the previous battle system interface we did user testing with 5 users, since it's considered to be a sufficient number of test users to find the most irregularities on an interface [17] [18].

Below, we present what was considered to be wrong with the interface, divided by categories based on what was described in section 2.6.2.
3.2.1 Inconsistency

Breaching consistency, we found the following usability issues:

• Action buttons have different sizes and not all have icons when they are together and represent equally relevant options (Figure 3.5).

![Figure 3.5: Some examples of usability issues found.](image)

• Most of the game’s actions open on a separate window whereas the attack action opens on a section inside the window representing the base/army you are attacking (Figure 3.6).

• On most locations, including the section to select units for an attack, the cards representing units have an information button on the top right to obtain further information about that particular unit. However, on the section representing the defending units you do not have that button which means you can’t access those units’ information to better select your attacking group.

• Attacking has a breadcrumb section, unlike any other section, which is a break in consistency.

• The layout of the simulation screen, which should be similar to the others, is completely changed. The breadcrumb section disappears and there’s no longer an information section with action but-
**Figure 3.6:** Unlike most of game actions, attacking opens up a nested window.

tons to the right. Instead there is only one row with several action buttons, like going to a previous step or to proceed and attack, and there is another row with the battle report itself.

- When returning from a further step, the units you selected previously still appear as selected in the information section but are no longer selected on the unit selection section.

### 3.2.2 Cryptic

Causing the interface to be cryptic, we found the following usability issues:

- When attacking, the breadcrumb section is not at the top of the page. Instead, it is after the main action buttons and between the information and the unit selection sections and its design does not stand out which leads to most users being unaware of its existence. Also, you start on step 2 of this breadcrumb.

- Contrast between background and text is not good on the side information section, which causes users to disregard some of the information that it is presented there.

- Location icons on the ‘To’ and ‘From’ sections are too small to be perceivable and are, therefore, mostly ignored.

- On the unit selection section, clicking on an unselected unit selects the whole group and clicking on a selected unit deselects the whole group which is not a common or expected behaviour.
• It is not clear how to select a different strategy and that the strategy icons are clickable to that effect.

3.2.3 Inefficiency

Causing the player to take too many actions to reach his objective, we found the following usability issue:

• Adding a partial amount of a specific unit resorting only to the mouse is a very tedious process as the screen input vertical arrows are the only way to do it and they are very small and add very little amounts, 1 unit, compared to the average used units, 100 or more units. As keyboard is not required or used for most parts of the game, it’s use should not be forced on this section.

3.2.4 Error inducing

A good interface should prevent the player from committing mistakes. Regarding this, we found the following usability issues:

• When the player has no amount selected of a specific unit and even though selecting a negative amount of units is not allowed, the system keeps the bottom vertical arrow on the input clickable which causes the player to commit an unnecessary error, even though the system immediately corrects that number to 0.

3.2.5 Recognition rather than recall

Straining the players’ memory, we found the following usability issues:

• Units have no clear indication of what they are great against which makes it harder to select units when attacking. Instead the user has to be constantly checking the unit information window.

3.2.6 Not providing feedback

The player should always know what’s happening and to that effect the interface should give proper feedback. Breaching this guideline, we found the following usability issues:

• When the player introduces an invalid number when selecting the amount of units he wants, the system automatically corrects it and doesn’t give proper feedback.
3.3 Strengths and Weaknesses

No system is perfect and neither is this and so, here we will be describing this system’s strengths and weaknesses derived from surveys conducted on our players and testers that came to our offices and from players’ behaviour when using our battle system.

To start with, this is a simple battle system, with a straightforward flow and easy to understand, so new players don’t have to learn much to start using it. It then allows players to get better at using it with time as they become aware of small details.

It also allows attacks to be made quickly and effectively. This way players can focus on their macro strategy, managing diplomacy and building and positioning armies on the map, instead of having to spend too much time preparing each attack.

However, this battle system also gives defending players 3 minutes to change their defence or run away which, in case that player is on-line, it is a long time because, if he is on-line when you launch an attack on him, knowing what he had defending at the time you send your attack will most likely be pointless as that player will probably change his defence before those 3 minutes have passed.

Furthermore, as the battle system will only select an attacking unit on each wave to attack a defence unit, the attacker has little incentive to send all 5 units. For example, if the defence has 1 infantry group of anti-air and 4 armoured groups of anti-armoured and anti-navy units, the attacker can only send 1 air unit that is good versus armoured and, odds are that it will destroy a lot more that it will be destroyed, as there is 80% chance of it hitting a target to which it deals a lot more damage than it receives. For this to be fair, all units, both attacking and defending, should have a chance to attack an enemy unit, once every wave, causing the attacking unit, in the previous example, to be attacked every round by the anti-air unit and, therefore, being quickly destroyed.

Finally, the choice of target by the attacking units is purely random, which can lead to very unpredictable outcomes. If your units always attack the units they are best against, you will have a much more favourable outcome than if they attack the units they deal close to no damage to and are dealt a great deal of damage.

3.4 Conclusions

Having done usability testing on the previous battle system and discussed its strengths and weaknesses we can now decide what and how to improve it to provide a better experience for our players which, in the end, is the purpose of this work.

With this in mind, to improve this battle system we decided to make the interface significantly more consistent and a lot less cryptic so that players are much less likely to feel lost or to miss out on important information displayed on the interface. The system should also remain simple, effective and quick to use.
while making significant to have more units than the opponent and reduce the random factor to avoid so many unexpected outcomes.

Also, regarding the state of the art in similar games (section 2.5), we decided that the player should be able to strategically position his units on the battlefield and to give meaningful strategies to each unit, allowing him to overcome stronger enemies’ defences. To be able to strategically position units and to have them moving, we decided to have a battlefield similar to Fire Emblem and Advance Wars, which are very clear on their unit movement and attack as well as simple. As for meaningful strategies, Total War: Warhammer and XCOM 2 had some interesting ideas such attacking while moving away, exchange movement for extra defence or even unique special strategies for different units.
In this chapter we will be describing the new battle system we propose to replace the previous one, both its design and system architecture. Its intent is for players to feel like they have more control over how battles play out and to make them want to battle each other more often and more intensively. This battle system should also allow for the battles to evolve further into a Real-Time Strategy (RTS) experience, giving players the option to change battle strategies in the middle of a battle and not just at the start. Afterwards, we will be detailing how it was implemented, its evolutions between prototypes and the challenges that we had to overcome, as well as the user tests conducted and how they helped us improve the battle system.

4.1 Final Design

The new battle system will essentially follow the same flow as the previous one (Figure 4.1), having a first step to choose units (Figure 4.2) and a second step for the player to execute his strategy (Figure 4.3). However, these steps will increase in overall complexity and flexibility, allowing the player for a wider range of choices that he can make on each step. The interface will also be reworked to fit the features that will be introduced and the ones that will be adjusted as well as to improve on the usability issues detected on the previous battle system described in section 3.2.

With this new battle system, the battle will happen on a battlefield represented by an hexagonal grid of size 9x7. Each unit will occupy a single tile and each tile can only be occupied by a single unit.
at a time. Units will then have additional statistics such as movement and attack range and can only be placed on roughly half of the battlefield, half for the attacking side and half for the defending side. Movement is the amount of tiles that unit can move on each wave and the attack range is number of tiles away from the tile it is in that the unit can reach another and attack in that wave. On navy districts there are also water tiles which can only be occupied by navy units. As these units cannot move into
land tiles, they have a larger range than any other units so they can target most of the battlefield, with
the exception of the navy units that target other navy units which have global range (to compensate for
the battlefield’s shape and size).

Unit strategies will continue to exist, but some of these will be reworked and there will also be addi-
tional strategies available, not all being available for every unit. The strategies will now be:

- **Default**: Units will move towards and attack their chosen target until the battle is finished or they
die. Units will never move closer than what they have to in order to be able to attack.

- **Cautious**: Units will behave the same as with the Default strategy, but will retreat if their numbers
are reduced below 70%. When retreating, these units will not attack and will move back as far as
they can towards their side’s back-line.

- **Guard**: Units will guard their initial position by moving at most one tile away from that position. If
they have no target in range and if they are not at their initial position, they will attempt to move to
their initial position.

- **Hold Position**: Very similar to the Guard strategy but more strict, forcing units to stay in their initial
position at all times.

- **Ambush**: This strategy replaces the Flank on the previous battle system, as Flank would not
correctly represent what this strategy does. This strategy causes the units with this strategy to
appear in the target position 5 waves after the battle has started and immediately attack the closest
best target. This target position can be anywhere on the battlefield as long as it is a valid tile for
that type of unit.

As a player chooses to attack another player, instead of being presented solely with a window on
which he has the units selected and a section with the units available to attack with, he will be presented
with the battlefield itself having, on the right side, the enemy units on the positions specified by their
owner. Strategies from the defending units are not shown to the attacking player and units under strate-
gies that would allow them to be placed on the attacking side (such as Ambush) are positioned on their
mirrored position on the defending side, so the attacking player has to anticipate for any possibilities.

Overlapping the left side of the battlefield, the player has the section with a list of all the units he
has available and which allows him to select the amount of units he wants to send into battle, as well
as information on the limits of what he can send and what he has already selected. On the right side,
below the battlefield, the player can see which rewards he might earn or what he can lose whether he
wins or loses, as well as a button to proceed into positioning and selecting strategies for his units.

To select units the player clicks on the unit card he wants. Afterwards, a small window appears which
has 4 buttons each with different quantities and an input box, allowing player to easily select the amount
he wants using only the mouse or if he wants to be more specific, he can use the keyboard on the input box and directly specify the amount he wishes to select. The player can also remove all the quantity selected by clicking on a button right of the input box. After he selects the first amount of a specific unit, that unit is added to the selected units list and when he removes the selected amount, that unit is removed from the selected units list. When units are selected and removed, they are also automatically added or removed from the battlefield, respectively, on their side of the battlefield, on semi-random positions. Infantry units are placed on any tile in the front-most row, followed by armored and finally the air units are placed. Navy units are placed randomly on the navy tiles.

Once finished selecting units, the player can then move on the next step, by clicking on the bottom right side button, in order to move units to other positions and/or changing their strategy if he wishes to do so. Units whose positions and strategy the player can change have a green decal to distinguish them from the enemy units and so players can more easily understand these units can be clicked on. On the other hand, enemy units have a information icon next to them so the player can obtain further information about them.

Once clicked, any unit will show their attack range by partially colouring the affected tiles and will show the numeric values for both attack range and movement range above the respective unit. Enemy units have no other behaviour when clicked but the player’s chosen units, once clicked, also have two additional behaviours. First, all tiles where the player can place that unit are highlighted and the tile which the mouse is hovering also has an icon displayed on it indicating that is the tile which, if clicked, the unit will be moved to. If the player clicks on any tile that unit is moved to that position and deselected. Second, once clicked, a button to select available strategies appears next to the unit and if clicked, will be replaced by a vertical list of all strategies available for that unit. If any is selected by the player, the unit will either be deselected or the player will be instructed to take any further action, which is the case of the Ambush strategy, where the player can choose another position for that unit across the whole battlefield.

Support is no longer done by allied players, instead, players can allow other squadmates to use their units in battle. To select them when attacking, all a player has to do is click on the support units tab next to the unit selection section. Once selected, all the process is the same as if the units were the player’s and the player can change to and from this tab at any time he is selecting units. After the player attacks, these support units will be temporarily unavailable to their owners until the battle finishes, at which point, if they survive, they are returned to them.

Finally, the player can choose to attack using the current units and strategies he chose, by clicking the Attack button on the bottom right. The player can also choose to go back to choosing units by pressing a smaller button next to this one. In the previous battle system, the attacking player could simulate the outcome of the battle but, with the increase in complexity in this new system, we decided that it would...
not make sense to be able to simulate, specially because the attacking player doesn’t have access to strategies used by the defending units.

Figure 4.4: Section where the player can access previous battle reports.

The player can also access battles that have already been processed in order to watch what happened or obtain information like damage dealt, units lost and others (Figure 4.4). He can do this by accessing a battle report of battles he has been involved in or battles reports that have been shared by other players. On those, he can immediately obtain information like location of the battle, players involved and units used and lost. He can then choose to watch the battle or obtain more details about that battle.

When choosing to watch the battle, a new window opens with the battlefield and the battle video begins processing immediately (Figure 4.5). It reproduces each wave sequentially and seamlessly, starting with the positioning of the units until the battle was resolved. The player’s UI includes a top horizontal bar that contains information like the defending and attacking players, the location where the battle went down, the current wave being shown and a set of buttons that allow pausing the battle, resuming, fast-forwarding and skip to the last wave of the battle. The UI also includes two side vertical sections, one left and one right, that display the attacking and defending units, respectively. These sections contain representations for each unit that contain the unit’s name, its total amount, the state
of that unit in that wave of battle (if it is dead, retreated or how many waves are left for it to enter the battlefield) and, in case it is a support unit, its owner's avatar. In the end of the battle, additional information will be appended to the side sections, that shows the damage ratio and the total amount of units killed for each unit.

**Figure 4.5:** Frame of a battle replay video.

If the player chooses to see the details section of the battle, this will show the damage done by each player and percentage of surviving units, just like in the previous battle system’s reports.

### 4.2 Solution Architecture

Our solution has implementations for both server and client sides, as each present its unique challenges and need to be integrated for a whole complete experience. The server side is where all calculations are made and where every information exchanged between the player and the game is checked to see if it has not been corrupted. The client side is what the player interacts with and what sends requests for information to the server of what the player is attempting to execute, whether it is accessing a different section or executing an action like attacking or upgrading a base.

It was done following a Model-view-controller (MVC) architectural pattern [19] to allow a better cohesion between server and client and for easier reuse of coding when expanding the game to another platform. With MVC the player makes requests to a controller which executes actions on a model. This model contains all the information relevant and is in charge of retrieving data from databases which will then be passed onto a view that will display this information to the player. In this case, the controller
and model are developed in PHP while the views depend on the target platform which in this case, is
developed in HTML5, CSS3 and JavaScript.

4.2.1 Server Side

Battles themselves are processed here as well as anything that manipulates information in a perma-
nent way, meaning that everything other that mere visual updates on the player’s interface is processed
on a server. Everything done here is developed using PHP and database manipulation is done through
MySQL.

Battles are processed by a Cron that runs every minute. This Cron cycles every pending attacking
unit group, gets its target location and defending group, and then processes the battle accordingly
(Figure 4.6). Each battle is processed in a similar way, with small differences, depending on the target,
location and type (Conquer and Destroy have some extra steps that aren’t consider in regular battles).
All initial information, such as the chosen units and their respective amounts, is stored in a JSON saved
on a database when the battle is over in order to allow the reproduction of this battle later on by the
player. Note that a player only selects which units will participate in a battle and how will they behave
and, once a battle has started to be processed, players will not be able to have any additional input
towards the battle’s outcome.

Firstly, all units are processed to create unit objects so that they can be more easily manipulated and
the hexagonal tile grid that represents the battlefield is created. Then we process the starting position
of each unit and place them on the battlefield. We start off by placing all units that have a specified
position in that position, however, if a unit has a strategy that hides it until a later wave, it is not placed
on the battlefield. Then, if it doesn’t have a specific position set, which happens only for defending units,
it is placed on the first available default position which is the position furthest away from the opposing player’s side. This positioning of units is equivalent to being wave 0, which will be used to reproduce the battle later on to the player.

Secondly, the battle is processed in waves, on which all units have their turn to take action. Units closer to the enemy’s back-line generally take their turn first, however, there are some strategies that overrule this. For example, units with Guard and Hold Position strategies are always the last ones to attack on each wave, while units with the Ambush strategy attack immediately before those and all other units attack first, according to the first rule. Each unit behaves differently and can choose to move, attack, hold their ground or retreat, according to a specified set of rules defined by their strategies and, unlike the previous battle system, units don’t automatically retaliate when attacked. The system saves a list of all actions taken in each wave and saves them in a JSON that will be stored in a database so that it can be reproduced later if a player wants to watch a battle again. Action information stored includes the unit that action belongs to, its type - move, attack, retreat, position - and, depending on the action’s type it can store additional information like the action’s target unit or tile and the damage dealt. Lastly, the battle is finished and the outcome is processed and saved in a database, on which several steps are taken:

1. The system decides who is the winner based on units survived and damage dealt. If the attacking player has surviving units and the defending player doesn’t, the attacker wins, and vice-versa. In case both attacking and defending players have surviving units, the player who dealt more damage always wins.

2. If the attack was a Conquer or a Destroy and if the attacking player won the battle, special actions are taken. If it was the former, the base will now be under the control of the attacking player and all its benefits will be transferred from the defending player to the attacking player. If it was the latter, the base will be removed from the game, and all its benefits will be removed from the defending player, releasing that base’s slot in the map for any player to build a new base.

3. Applicable rewards and penalties are given to all intervening players such as experience, battle points and resources, depending on whether a player was on the winning or losing side.

4. Progress on missions and awards are processed for all intervening players.

5. Battle information is stored in database, such as units’ actions on each wave and the battle outcome.

6. Notifications are sent to all intervening players that the battle has been processed and that they can access it.
4.2.2 Client Side

On the client side there are two different interfaces. The one where you prepare your defence/attack and the one which allows the player to watch previous battles. Both interfaces are developed with JavaScript to maintain most processing on the client side, avoiding delays in response when interacting with the interface caused by distance from the server, as this is a game to be played globally.

The interface where the player prepares his defence/attack, unlike other sections in the game, is processed mostly on the client side, with only a few server calls to increase coherence. After receiving the initial information required to create the battlefield and build all interface elements from the server, all selections made by the player are stored in a JSON as the value of an HTML5 input tag. There are two different input tags used in this whole process, one for attacking player’s units and one for his selected support units. The JSON contains information such as the identifier of the unit, the amount chosen, the position selected on the battlefield and the strategy identifier.

Selecting an amount for a specific unit, positioning it or changing its strategy, will simply modify this JSON and all this information is manipulated client side which means that the player doesn’t have to wait for the server to give back information. The only times the client has to wait for a server request is when he goes back to editing selected unit amounts or changes between selecting his own units or support ones, as an effort to update the information displayed to the player in order to make it as coherent as possible, since other players might have taken action after you started an attack or even you might have been attacked, which does not happen often but there is always a chance for it to happen.

On the other hand, in order to watch a battle that has already been processed, the client makes a request to the server for the list of actions taken on each wave, which were previously stored in a JSON on a database. Using this, the system can reproduce exactly how the battle happened and obtain the same result without having to store units statistics states every time an action is taken, instead it creates the unit objects on the client with the initial information that was also stored in a JSON on a database. So, with the units having been created with the correct amounts, the client only has to process each action on each wave and update the involved units’ placement and status, achieving a coherent battle replay, storing less information on a database.

4.3 Implementation and Testing

The battle system described in the previous sections wasn’t the one that was firstly implemented, as this was an iterative process during which we made some tests with players to find out which features and interface issues could be improved. Most of these changes were made regarding UI but there were also some changes that had to be made regarding the system itself.

The first version of the battle engine was tested by players in events. Afterwards, we worked out a
few issues and introduced the new battle engine in the game as a side challenge, unrelated to the game itself, to provide additional feedback from the core players of our game. Finally, we replaced the previous battle system with the new one. This allowed us to receive feedback from the players about what they liked or disliked most about it and how well could they use it. We gathered this information both through surveys and through our own metrics, which allowed us to know which units were chosen, where they were positioned and if any of these decisions had impact.

4.3.1 Version 1

The first version of the new battle engine was very simple and was tested on two public events where people came to our booth and played it. In both the public events and in our offices, we had two types of players test our system, those new to strategy games and those who were already familiar with strategy games’ mechanics. We presented the players with a challenge which consisted on overcoming as many levels as possible, each with a different predefined defence that grew increasingly more difficult with each level passed. The players would be able to choose from all of the available units in the game, both normal and supreme, and would have a limited budget and a limited number of different units they could choose, just like when playing the actual game. To play this challenge the player was not required to create an account, thus enabling us to get users to test the battle system anywhere.

This prototype was first tested at Web Summit, a huge technology conference, where we had a small booth with the challenge ready for players to play. We had 67 people play our challenge but, as this was a huge event, most players didn’t play for long and therefore didn’t have time to get used to the mechanics but even these players who didn’t stay for long enabled us to get feedback regarding how players feel when they first get in contact with the new battle system. We also managed to get a few players to play a little bit longer and provide us with a little more insight towards how players feel after playing for a while. A few weeks later we did a similar test at Cogito, a much smaller event whose audience was not very interested in games. Despite this, we managed to have 23 people play our challenge, mostly inexperienced players of strategy games.

In these events, we explained the players what the game was about and gave them a general idea of what they would be doing to contextualize them and provide them a more enjoyable experience. From that moment on, we would take notes of what they were doing and what they were struggling with, but we would only intervene and help them if they were lost on what to do, in which case we would point them in the right direction and let them continue playing.

Players started out by looking at each unit and using the information button on the unit card to obtain further information about them, this would allow them to know which units might be useful against the enemy defence.

Picking units and the selected unit section behaved very similarly to the final version, although the
interface was a bit less polished in some aspects. Afterwards the player had to click on the bottom right button to proceed to placing the units on the battlefield, similarly to what he has to do now. However, unlike the final version on which the units selected are automatically placed on the battlefield, in this version the player was forced to place the units on the battlefield one by one.

After placing the units, if the players wished to proceed he could just click on the attack button that was in the same place as the button to position units but if he did not, the only way to go back to choosing units and their respective amounts was to click a side tab on the selected units section.

As for the battle processing itself, the unit pathing and targeting algorithms started by being fairly simple but it soon became clear that they had to be tweaked to meet the players' needs. The pathing algorithm started by being a simple A* algorithm that would find the closest path leading to the closest hexagon near the target. This pathing algorithm did not take into account that you should be able to overcome an obstacle if your movement range was higher than the obstacle and if there was an available position beyond the obstacle. On the other hand, the targeting algorithm started by being as simple as any unit would target the unit which they would deal the most damage to, even if that unit would have little impact on the fight.

In order to obtain better results from player testing, we realized that our challenge should be modified in some aspects. For instance, players could overcome most levels with supreme units, without having to give much thought as to which units to use since supreme units are much stronger than normal units and used a separate budget that could be used together with the one for normal units, allowing the player to select too many units. Also, since in this challenge you are not using your own units as opposed as if you were playing the actual game, supreme units are too strong since they have no cost and no risk of loss.

After some minor changes, we integrated the challenge on the game itself, as a side challenge and not replacing the previous battle system yet, which would allow us to obtain direct feedback from our core players who actually play our game on a daily basis. Most of our active players played our challenge, which allowed us to obtain data and feedback from both hardcore and casual players. We asked them to fill out a survey to allow them to express their feelings about the new battle system, whether they liked it or not and what did they think was missing or poorly accomplished. 60.4% thought the new battle system was above average while only 14.5% though it was below average and 72.9% said that they would like this new battle system to replace the previous one. They also gave us a lot of feedback about which strategies they would like to see implemented in the game, as well as pointed out several bugs and missing information which made it hard for them to plan their strategy.

All this feedback allowed us to better prioritize the changes that were needed on the battle system to improve player experience both on initial contact and on prolonged playing.
4.3.1.1 Usability analysis

Between the players in events and the players accessing the challenge from inside the game, we were able to identify some usability problems and to figure out what worked well and what caused confusion to the players.

For starters, we realized that having all units to choose from made players feel confused and frustrated as it took a lot of time to process all the information and by the time they had looked at 10 different units they started to forget what the first ones were good and bad against which lead to an erratic behaviour of going back and forth between units. Plus, when they successfully passed a level, this behaviour would start all over again, because the defence was made of a different set of units and therefore the player needed to obtain different information than last level. On top of all this, the player had no way of obtaining any information about the enemy defending units, which caused even more confusion to the players, some of which even tried to match the units on the battlefield to the unit cards they had to choose in order to obtain some information.

When selecting units, players felt overall comfortable although the input box within the selection box did not allow clearing the total amount selected of a specific unit through the single click of a button which players considered to be a better way to remove the amount selected for a specific unit.

The next step caused a lot of confusion because, although the unit being placed was highlighted on the selected units section, players did not notice it at first and when they did they tried to drag the unit card onto the battlefield in an attempt to position the unit on the battlefield. This was mostly due to the lack of visual feedback on the tile selected by the mouse, which should show a reference that the player was placing a unit there, for example, a faded image of the unit. Finally, when placing the units, players did not know how far the units were able to move on each turn or how long was their range of attack and so they placed the units randomly, only getting aware of some of these statistics after playing a few levels as this information was not available anywhere.

Another problem with the interface was the way to go back to selecting units from the deploying units step. As stated, the only way to do this was to click on a side tab on the selected units section, which was not clear for most players.

As for unit movement, units always tried to get as close as possible to their target even if they could target the enemy unit from a farther distance, which led to units clustering together near the middle of the battlefield and caused players to lose interest in the battle. Also, units with high movement range (2 tiles or more) were not able to jump over others which, for the players, was not the expected behaviour since most of these units were aircrafts.

Finally, considering the units’ targeting algorithm and the fact that units simply targeted the unit they would deal more damage to, we found that, although predictable, players felt that this behaviour was not very intelligent and that it compromised the strategic factor of battles, as it caused players to lose battles.
they thought would be an certain victory.

### 4.3.2 Version 2

After analysing the results obtained from the first prototype, we made adjustments towards improving the player's experience in order to introduce the new battle system as the game's main battle system.

We started by adding a way to change units' strategy, and developed the strategies themselves. Players would now be able to change their selected units' strategy after placing them on the battlefield. Once placed on the battlefield, players could click the unit and a side button would appear, which would lead to a menu listing all available strategies for that unit. The player would then click on the strategy he wanted for that unit and that menu would disappear and the unit would be deselected. As for the strategies available, we started by developing these four strategies: *Default*, *Cautious*, *To the End* and *Hold Position*. These were already very similar to their final versions as described in section 4.1 except for the *Default* strategy with which units would retreat if their numbers were reduced below 30% and the *To the End* strategy which worked exactly as the final *Default* strategy and therefore would end up becoming the *Default* strategy.

After adding strategies to the battle system and taking advantage of the our game's new season start, we replaced the previous battle system by this new one on the game itself. By doing so, we would get faster feedback from the players that were already playing the game using the previous system and that could provide better insight in some aspects such as unit balancing and overall usability. In addition to having our players use our new system, we would still have other players test our battle engine both in our offices and in events for a more complete usability testing.

Supporting was not initially allowed so we could better address problems that might arise when launching a new feature, also derived from multiple people interacting with each other at the same time. After everything was considered stable, we introduced supporting as described in 4.1.

We also added the *Ambush* strategy shortly after launching this system. This strategy caused the unit to enter the battlefield five waves after the battle had started and on the position that mirrored the initially selected one. By mirrored we mean the position equivalent as if we flipped the battlefield.

This prototype version was not tested in events or by presential player testing and feedback but was, instead, very frequently iterated on based on direct player feedback from the players that use it on a daily basis. This way we were able to address it's main issues faster and more effectively. As the new battle system was now integrated in the game, every bug that came up needed to be resolved immediately and every feature that was flawed need to be reworked as soon as possible to keep the players engaged and having an enjoyable experience.
4.3.2.A Usability analysis

Most usability problems found in the previous prototype version persisted although some of them lost some of their impact. Deploying units on the battlefield and returning from this step back to selecting unit amounts remained acknowledged as a confusing step but the lack of information on defending units was found to have less impact than before, because in the game players are slowly introduced to each unit and have time to familiarize with them and learn to identify them on the battlefield. The pathing and targeting algorithms were still considered flawed and the lack of information about units’ movement and attack ranges was also considered a critical problem.

By observing players both through our game or playing our challenge, we realized that the Default strategy should not make units retreat. When units were retreating they would not attack other units and would, therefore, be vulnerable until they retreated. Because of this, most of them would die anyway which caused inexperienced players to lose more units and battles than they should and experienced ones to always use another strategy, such as the To the End, which caused the units to not retreat.

In addition, most people had trouble finding how to use strategies as some wouldn’t even realize that the units could be clicked and others didn’t realize that a button appeared next to the unit whenever you clicked it. Furthermore, the Ambush strategy was not clear on how it worked, as most players selected the position they thought the unit would appear in and then selected the Ambush strategy. Others just thought it appeared on a random position as there was no visual feedback on where the unit was going to appear. The Hold Position strategy was also found to be flawed, as units with shorter attack range would just stand by and get attacked without retaliating if the enemy unit had more attack range or let other friendly units close to them get attacked without helping if the enemy was out of range.

Finally, by having separate budgets for supreme and normal units, players which didn’t have many supreme units were often confused on how they could overcome some enemy armies and felt that the new system was very unfair if one side had much more supreme units than the other, as players with both supreme and normal units would have basically double the budget to spend. Also, as a player can only send 5 different types of units, players with 2 or 3 supreme units could send a lot of normal units which were always targeted before the supreme units because they took more damage, allowing the supreme units to destroy everything untouched.

4.3.3 Final Version

After having tested the system more thoroughly with new and old players of our game we could then make more modifications to the battle system so that players could have the best experience possible, either being experienced players or not.

Firstly, we made improvements to unit deployment step, both in UI and behaviour. Instead of forcing
the player to position the units on the position they wanted, the units were placed automatically on a semi-random position each time the player selected an amount for a specific unit on the previous step, as described in 4.1. Furthermore, the units that the player owns on the battlefield are now highlighted to better indicate that some action can be taken by clicking on them. Finally, we added an information icon next to every enemy units, so that the player can obtain more information about those particular units, and clicking on any unit in the battlefield will now visually show the attack range of the selected unit and will display a bubble that indicates numerically both movement and attack ranges. We also removed the side tab that used to allow going back to the unit selection step and added an orange smaller button with a left arrow icon next to the attack button, to that end.

Secondly, on the first step, to select unit amounts, to balance and simplify the unit amount selection, we merged the supreme unit budget with the normal unit budget and adjusted the overall unit cost towards this budget, having supreme units cost a lot more than normal units. Also, to allow clearing the previous selected amount for a specific unit, we added a button next to the input box on the section presented after the player clicks on a unit card.

Thirdly, we made some modifications regarding strategies. We replaced the Default strategy with the To the End strategy and removed the To the End strategy, effectively reducing the number of available strategies by one. Then, we adjusted how the Ambush strategy was set up so it was clearer to players. When choosing the Ambush strategy players would now be able to select the target position from any available on the battlefield instead of choosing a position on their side and then that position being mirrored to the other side. We also added an additional strategy to replace the Hold Position strategy, in most cases, called Guard. This would work similarly to the Hold Position but would allow the unit minor movement, as described in 4.1.

Finally, we adjusted the movement and targeting algorithms. The pathing algorithm remained very similar to what it was but we modified it so that the units would only traverse the shortest path towards their target and would stop on the furthest position that would allow them to attack and not any closer. The units were now also able to overcome other units or obstacles, provided that they had the movement range necessary to do so. As for the targeting algorithm, units would now take in consideration 4 steps, Threat, Damage, Distance and Time To Kill. Threat would have a weight of 0-10, Damage 0-10, Distance 0-infinite and Time To Kill would only have a weight of 0-3 as it was intended merely as a tie break. On the Threat step, the unit being processed would consider, for every enemy unit, the damage potential to the its own team. This damage potential was an average of the possible damage dealt to each unit of said team. The Damage step was the same as in version one, where the unit being processed would check to which unit would it deal the most damage. The Distance step would calculate how many turns would its movement + fire ranges take to being able to fire on a chosen target. This value was exponential as the longer a unit takes to reach its target the less it is maximizing its potential for each
Finally, the Time to Kill step would check, between two closely possible targets, if a unit would take fewer turns to destroy one over the other and choose the one it would destroy first, as to minimize the opponent damage. No step in the targeting algorithm would take in consideration enemy strategies or expected behaviours.

This final version was tested by players in events, players that came to our offices and, of course, by the players of WWO which used this final version on a daily basis and submitted feedback often to tell us about bugs or features they would like to see changed and/or implemented. As for events, we tested this version of the battle system adapted in a similar challenge as the one used on the previous events we had performed testing. First, we tested our game at Jornadas InfoWeb, a small university event, on which we managed to have 33 players to play our challenge, mostly gamers and with players both new and familiar with strategy games. Then, we tested our game in a larger event, AEIST JobShop, also in a university, where we had 79 people playing our challenge. As for testers that came to our offices, we had 12 players which tested the game in different times, following minor changes to the game, allowing them to find different usability issues.

4.3.3.A Usability Analysis

Most of the previously detected usability issues were corrected but players still found confusing some parts of the attacking process.

As the player’s side of the battlefield is overlapped by the unit selection section, players didn’t realize that the units were automatically placed on the battlefield which caused confusion when they proceeded to the stage of moving units and selecting strategies. As the units were already placed on random positions, from the player’s point of view these positions were final and it was not obvious that action could be taken by clicking on them. The highlight which was added on the player’s units helped but this problem still persisted.

The attack range information we added when the player clicked on a unit was also not very clear, as some players thought that the tiles that became highlighted were the range to which they could move the unit while preparing the attack.

4.3.4 Conclusions

Even though this final prototype was much better accepted by both experienced and new players there were still a few issues that remained to be fixed. Some of these problems were reported more often by inexperienced players who did not have the system explained to them through a tutorial which means that the system itself should still be clearer on what the user can and cannot do.

On the other hand, players thought that these new battle system brought a more strategic approach to battles and that enabled battles to be more interesting to play and to watch.
Final Results and Conclusion

Contents

5.1 Data discussion ......................................................... 55
5.2 Conclusion ................................................................. 57
5.3 Future Work ................................................................. 57
5.1 Data discussion

In order to evaluate the impact of the new battle system, we analysed the number of battles fought, the damage dealt and the number of units killed, per day, for a whole season before the new battle system was integrated in the game and a whole season after it was. We compared two whole seasons because during a season there are natural fluctuations in these numbers that are common to any season, such as a much lower battling activity in the first few days of the season.

Once the new battle system replaced the previous one, we noticed a slight decrease in the number of battles fought by players, as shown in Figure 5.1. However, we noticed a large increase in damage dealt and units killed per battle, as shown in Figure 5.2 and Figure 5.3, respectively. This meant that, although players were fighting less, the battles they fought were a lot bigger and meaningful and players felt that their choices had a much bigger impact on the outcome of each battle. However, when considering defending, players still thought that it was really hard to win on defence since there was no way to know what you were going to be up against and it was also impossible to make adjustments to your defence setup after the attack had happened. Despite this, as positioning and strategies had a more meaningful impact on the battle and because the attacker had no way of knowing the defender's strategies, using these effectively on defence allowed the player a better chance at winning.

Regarding the number of players playing the game, we didn’t notice much change. This might have been influenced by the fact that some game rules, such as capital conquering and resource stealing rules, were changed close to the time the battle system was replaced. Therefore, we could not obtain a number that would correctly indicate how many more players joined the game because of the introduction of the new battle system. According to the opinions from new players which had tried both battle engines, obtained through surveys and personal contact, players found that this new battle system was much
more appealing than the old one and that made them curious in trying it out. After trying it out, they felt interested and engaged as they could make several choices that would impact the outcome of the battle.

Overall, we felt that the new battle system had a positive impact on the game and that both old and new players prefer this new battle system to the old one, despite not being able to assert how much of the growth in player number or player activity in the game was due to the new battle system. Every time a change in the game happens there is always a chance that some players will play less while adjusting to that change or that a player will temporarily or permanently stop playing the game and, in a game that is always adapting and evolving, sometimes it is hard to say with certainty how much impact did a change have on the game.
5.2 Conclusion

The results obtained from this work were very positive, as we were able to reach our goals. Players felt that this new battle system gave them a better sense of control and that they had the opportunity to make choices that mattered for the outcome of the battle. We also aimed to have a battle system which would allow the player to give inputs in different stages of the battle for a more dynamic experience but, although the system itself is prepared to take this in consideration, there were latency issues and overall game dynamics which in the end forced us to postpone this idea.

As for player activity, we did not notice any relevant shift in activity, neither positive nor negative, and, although we could not determine for sure how many players joined the game because of the new battle system, it was specially important that there was not a noticeable decrease. It was important because, when changes as big as this one are introduced, some players feel resistance to change and sometimes end up leaving the game, even if temporarily. We were also able to make our battles much more interesting for players and to have players much more engaged in them, having had an increase on average damage dealt and units killed by battle and by day and also having our players tell us that the new battle system allowed them to have much more meaningful choices when preparing an attack in order to overcome a stronger foe.

It is also important to notice that the game itself suffered some changes unrelated to the battle system itself during this work, like changing some capital conquering and resource stealing rules, which might have influenced some of the results obtained.

5.3 Future Work

In the future it would be nice to have players playing more dynamically against each other, being able to give input during battle to change how their units behave or even add or change other units.

It would also be nice to have a wider variety of strategies available for each unit, as well as having strategies unique to each unit allowing each unit to have a special role in each battle.
Bibliography


