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

João Luís Parracho Fazenda dos Reis
joao.l.reis@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2017

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. Finally, we present the user testing methods and respective results obtained for this new battle system.
Keywords: Battle system, strategy game, user interface, user testing

1. Introduction

This thesis was integrated in a company, Chilltime, whose main product is World War Online (WWO). WWO is a Massive Multiplayer Online Game (MMORTS), currently browser based, built using mainly PHP and Javascript. It's similar to several established games, such as the Total War series [3], GoodGame Empire[17] and Forge of Empires[7], in terms of overall concept, game mechanics and/or interface.

This work focuses mostly on the battle system and, with it, 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.

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.
3. increase the number of battles made per player by 10%;
4. increase the number of players actively playing our game by 25%;

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.

2. Game Overview

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 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.

As for gameplay, the player can move around a world map (Figure 1), 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 champi-

onship. 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.

Figure 1: Map View of Europe.

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. 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.

3. State of the art in similar games

In order to find the best possible solution to improve WWO's battle system, we analysed several games that fit in the same genre as WWO and/or shared similar features. There were several games that fit this criteria so we focused more on the ones that had a similar target audience and the ones which shared a similar gameplay or a strategic battle system. These games were Tribal Wars 2 [8], GoodGame Empire [17], Command & Conquer Tiberium Alliances [6], OGame [5], Forge of Empires [7], Endless Legend [16], Total War: Warhammer [3], Planar Conquest [9], XCOM 2 [1], Fire Emblem [13] and Advance Wars [18].

From this analysis we were able to better define our battlefield layout, key actions, unit strategies and overall system dynamics.

4. Usability in Games

User Interface usability is critical for a game to succeed. 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 [14]. Therefore it is of the utmost importance to test with real potential users. 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 [4].

4.1. User Interface

An UI is what allows players to take action on a video 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 [14].

As stated in [14], an interface is considered dysfunctional when it is cryptic, complex, simplistic, inconsistent, inefficient and/or cluttered.

To create a great game interface, there are also several guidelines suggested by many authors [14] [2] [10].

4.2. 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 [15].

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.

4.3. 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, and verify if players are using the interface as expected.

5. Previous battle system

5.1. 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.

When a player decides to attack another, he is first 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.

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 . These strategies can influence the unit's behaviour in battle and/or change their statistics slightly.

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. 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 1).

In the end, after the battle has been processed, all players involved receive a battle report detailing what happened in the battle. On each battle report the player can see the gains/losses for each side and he can also 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 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.

begin
  $maxWaves \longleftarrow 50$
  $currentWave \longleftarrow 1$
  $unitListDef \longleftarrow getUnitListDef()$
  $unitListAtk \longleftarrow getUnitListAtk()$
  while $currentWave <= maxWaves$ **do**
    foreach $unit \in unitListAtk$ **do**
      $targetUnitDef \longleftarrow$
       $getRandomUnitDef(unitListDef)$
      $attackTargetUnit(targetUnitDef)$
      $retaliateDefUnit(targetUnitDef, unit)$
    end
    $prepareNextWave()$
  end
end

Algorithm 1: Battle Processing Main Flow

5.2. 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 [11] [12].

From this testing we were able to identify 15 usability issues related to inconsistency, crypticness, inefficiency, inducing user error, recognition rather than recall and lack of feedback.

5.3. Strengths and Weaknesses

To analyse what players felt when using this battle system, we conducted surveys on our players, had testers come to our offices and analyse player behaviour derived from our metrics.

We concluded that this was a simple and easy to use battle system, allowing players to focus on their macro strategy and not blocking new players, but which then had very unpredictable outcomes because unit targeting was too random, making players feel like their choices when attacking did not matter.

5.4. Conclusions

Considering the usability testing and this battle system's strengths and weaknesses, 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 3), 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.

6. Final Design

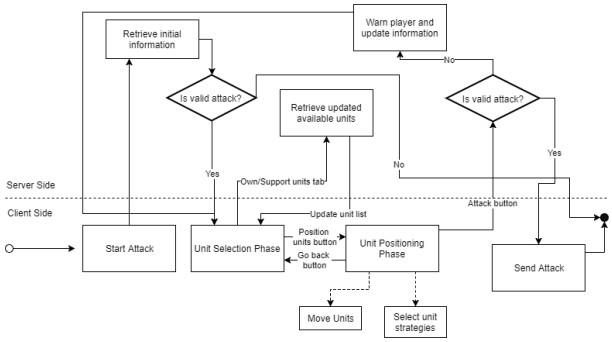

Figure 2: Sequence of steps required to make an attack.

The new battle system will essentially follow the same flow as the previous one (Figure 2), having a first step to choose units (Figure 3) and a second step for the player to execute his strategy (Figure 4). 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 5.2.

Figure 3: User Interface after choosing to attack a target.

Figure 4: User Interface for positioning units on the battlefield after selecting units to attack.

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 additional 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 strategies 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 frontmost 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.

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. 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 5).

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.

7. 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 com-

Figure 5: Frame of a battle replay video.

plete 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. 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.

7.1. Server Side

Battles themselves are processed here as well as anything that manipulates information in a permanent 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.

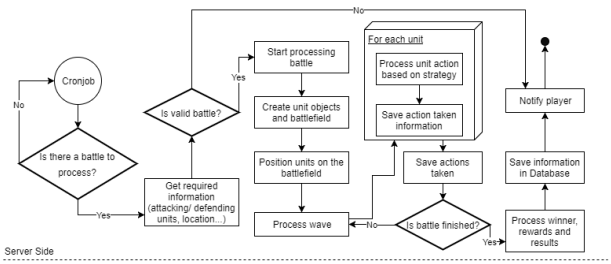

Figure 6: Sequence of steps taken when processing a battle.

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 6). 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.

Firstly, all units are processed and the battlefield initialized. Secondly, the battle is processed in waves, on which all units have their turn to take action. 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. Lastly, the battle is finished and the outcome is processed and saved in a database.

7.2. 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.

Selecting an amount for a specific unit, positioning it or changing its strategy, will simply modify a 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.

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 and the initial information. Using this, the system can reproduce exactly how the battle happened by executing the actions on the initialized units.

8. 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.

8.1. 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.

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 and we were able to get feedback regarding how players feel when they first get in contact with the new battle system and 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, 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.

Picking units and the selected unit section behaved very similarly to the final version, 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.

As for the battle processing itself, the unit pathing and targeting algorithms started by being fairly simple. The pathing algorithm started by being a simple A* algorithm that would find the closest path leading to the closest hexagon near the target. 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.

Then, 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.

8.1.1 Usability analysis

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. On top of 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 players did not understand how they could deploy a unit on the battlefield, 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. 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.

8.2. Version 2

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, similarly to how they do it in the final version. As for the strategies available, we started by developing these five strategies: *Default*, *Cautious*, *To the End*, *Hold Position* and *Ambush*. These were already very similar to their final versions as described in section 6 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*.

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, and was later added as described in 6.

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.

8.2.1 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 usually die anyway.

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 and 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 felt that the new system was very unfair, as players with both *Supreme* and *Normal* units would have basically double the budget to spend. Also, *Normal* units were always targeted before the *Supreme* ones because they took more damage, allowing the *Supreme* units to destroy everything untouched.

8.3. Final Version

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, as described in 6. 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. 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. We also added an additional strategy to replace the *Hold Position* strategy, in most cases, called *Guard*, as described in 6.

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. 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. 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 Job-Shop*, 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.

8.3.1 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.

8.4. 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.

9. Final Results

Once the new battle system replaced the previous one, we noticed a slight increase in the number of battles fought by players. However, as capital conquering and resource stealing rules were changed close to the time the battle system was replaced, we could not obtain a number that would correctly indicate how much more often did players engage in battle due to the new battle system. Players also 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 noticed a slight increase, although similarly to the increase in the number of battles fought by players, we could not obtain a correct number that would indicate how many more players joined the game. 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 and, after trying it out, it felt interesting and engaging as you 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.

10. Conclusion

Although some objectives weren't reached, the results obtained from this work were very positive. We were able to increase the number of players actively playing our game 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 an increase, no matter how slight. We are 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 battles made by each player 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.

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.

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.

10.1. 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.

References

[1] 2K. Xcom 2. Accessed 10-August-2017.

[2] E. Adams. *Fundamentals of Game Design*. New Riders, third edition, 2014.

[3] C. Assembly. Total war: Warhammer. Accessed 10-August-2017.

[4] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*. Pearson Education Limited, third edition, 2004.

[5] Gameforge. Ogame. Accessed 10-August-2017.

[6] E. A. Inc. Command & conquer: Tiberium alliances. Accessed 10-August-2017.

[7] InnoGames. Forge of empires. Accessed 10-August-2017.

[8] InnoGames. Tribal wars 2. Accessed 10-August-2017.

[9] W. Interactive. Planar conques. Accessed 10-August-2017.

[10] J. Nielsen. 10 usability heuristics for user interface design. *Nielsen Norman Group*, 1995.

[11] J. Nielsen. How to conduct a heuristic evaluation. *Nielsen Norman Group*, 1995.

[12] J. Nielsen. Why you only need to test with 5 users. *Nielsen Norman Group*, 2000.

[13] Nintendo. Fire emblem. Accessed 10-August-2017.

[14] J. Novak. *Game Development Essentials: An Introduction*. Delmar, Cengage Learning, third edition, 2012.

[15] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: beyond human-computer interaction*. Wiley, fourth edition, 2015.

[16] A. Studios. Endless legend. Accessed 10-August-2017.

[17] G. Studios. Goodgame empire. Accessed 10-August-2017.

[18] I. Systems. Advance wars. Accessed 10-August-2017.

[19] Wikipedia. Model–view–controller. Accessed 10-August-2017.