Ancient and Medieval Battle Simulator

Pedro Almeida d’Eça Moraes Vaz

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Júri

Presidente:
Orientador:  Prof. Pedro Alexandre Simoes dos Santos
            Prof. Rui Filipe Fernandes Prada

Vogais:

Setembro de 2010
Abstract

The objective of this work is to develop a platform that allows players to play the role of a general in ancient and medieval battles. Players create tactical plans describing how their armies will be placed in the battlefield, how they will behave, and then watch the battles in which their armies participated. This platform consists of a Tactical Planner, a Battle Simulator and a Battle Viewer.

The Tactical Planner is a Graphical User Interface (GUI) that allows players to create rich and varied tactical plans with ease. In each tactical plan, players can decide what types of soldiers to use, where to place them in the formation, and give them orders to follow during battle. The tactical plan is independent from the specific composition of the army.

The Battle Simulator is a MultiAgent System that simulates battles, using the formations and orders defined in each tactical plan and two given armies. Every contingent (group of soldiers in an army) will be played by an autonomous agent that will follow its orders. Each battle is stored so players can watch it in the Battle Viewer.

The Battle Viewer is a GUI that shows players a battle of their choosing, which has already been simulated, together with some statistics about that battle. The Battle Viewer should allow players to understand, just from watching the battle, if they made the right choices with their tactical plan, or if they have to improve it for future battles.

Keywords

Ancient, Medieval, Tactics, Battle, Simulation
Resumo

O objective deste trabalho é desenvolver uma plataforma que permita aos jogadores fazerem o papel de um general em batalhas da antiguidade e medievais. Os jogadores criam planos tácticos que descrevem como os seus exércitos vão ser colocados no campo de batalha, como se vão comportar, e depois vêem as batalhas em que os seus exércitos participaram. Esta plataforma consiste num Planeador Táctico, num Simulador de Batalhas e num Visualizador de Batalhas.

O Planeador Táctico é uma Interface Pessoa Máquina (IPM) que permite aos jogadores criarem planos tácticos ricos e variados com facilidade. Em cada planeamento táctico, os jogadores podem decidir que tipos de soldados vão usar, onde os vão colocar na formação, e que ordens lhes vão dar para executarem durante a batalha. O planeamento táctico é independente da composição de qualquer exército.

O Simulador de Batalhas é um Sistema MultiAgente que simula batalhas, usando as formações e as ordens definidas em cada planeamento táctico, mais um exército para cada lado. Cada contingente (grupo de soldados num exército) será controlado por um agente autónomo que seguirá as suas ordens. Cada batalha é guardada para os jogadores as poderem rever no Visualizador de Batalhas.

O Visualizador de Batalhas é uma IPM que permite aos jogadores verem uma batalha à sua escolha, que já tenha sido simulada, juntamente com algumas estatísticas sobre essa batalha. O Visualizador deverá ajudar os jogadores a compreenderem se fizeram as escolhas certas no seu planeamento táctico, para uma determinada batalha, ou se têm de melhorar esse planeamento para batalhas futuras.

Palavras-chave

Antiguidade, Medieval, Táctica, Batalha, Simulação
Acknowledgements

I would like to thank: professor Pedro Santos, for his enthusiasm in explaining the course of some ancient and medieval battles from our History, for helping me organize my thoughts and for his patience in answering all my questions; Marco Quinta, for complementing professor Pedro's knowledge about ancient and medieval battles; professor Rui Prada, for always going straight to the point and for being practical; Marco Vala, for his fresh ideas that helped us improve the Tactical Planner; professor Manuel Fonseca, for the thesis he provided, which helped me organize mine; all the subjects who helped improve the Tactical Planner by testing the prototypes; all the subjects who tested the platform in its final version; and last but not least, my parents, because without them I would not have reached this point in my life. Thank you all!
# Table of Contents

Abstract........................................................................................................................................... ii  
Keywords.......................................................................................................................................... ii  
Resumo ............................................................................................................................................. iii  
Palavras-chave .................................................................................................................................... iii 
Acknowledgements........................................................................................................................ iv  
Table of Contents ........................................................................................................................... v  
Figures .............................................................................................................................................. vii  
Tables ............................................................................................................................................... viii  
Acronyms ......................................................................................................................................... viii  
Definitions ......................................................................................................................................... ix  
1 Introduction................................................................................................................................... 1  
  1.1 Problem..................................................................................................................................... 1  
  1.2 The Battle of Cannae .............................................................................................................. 2  
  1.3 Objectives .................................................................................................................................. 4  
  1.4 Dissertation Layout ................................................................................................................ 5  
2 State of the Art............................................................................................................................... 6  
  2.1 Placing an army on the battlefield .......................................................................................... 7  
  2.2 Shapes, combat options and terrain ....................................................................................... 12  
  2.3 Available orders for the army .................................................................................................. 14  
  2.4 Battle status ............................................................................................................................ 16  
  2.5 Movement on the battlefield ................................................................................................. 20  
  2.6 Event-driven behavior ............................................................................................................. 22  
  2.7 Comparing solutions ............................................................................................................... 23  
3 Conceptual Model....................................................................................................................... 29  
  3.1 Concepts.................................................................................................................................... 29  
  3.2 The Battle of Cannae in our Conceptual Model .................................................................... 36  
  3.3 Platform Architecture ............................................................................................................ 39  
4 Tactical Planner............................................................................................................................ 41  
  4.1 Low-Fidelity Prototypes........................................................................................................... 42  
  4.2 Functional Prototypes ............................................................................................................. 50  
5 Battle Simulator and Battle Viewer............................................................................................. 57  
  5.1 Battlefield: The Environment ................................................................................................ 57  
  5.2 Placing contingents in the battlefield .................................................................................... 58  
  5.3 Battle Contingent: The Agent ................................................................................................ 64  
  5.4 The Battle Simulator .............................................................................................................. 71  
  5.5 The Battle Viewer .................................................................................................................... 72  
6 Playtesting the platform............................................................................................................... 77
Figures

Fig. 1 – Initial formations and orders in the Battle of Cannae
Fig. 2 – The Roman army completely surrounded in the Battle of Cannae
Fig. 3 – Placing the troops in a Total War battlefield
Fig. 4 – Placing the troops in a Spartan battlefield
Fig. 5 – Troops already placed in an Alexander battlefield
Fig. 6 – Troops already placed in a Centurion battlefield, according to the player’s formation
Fig. 7 – Placing the troops in a Legacy battlefield
Fig. 8 – Troops placed in a DBM battlefield
Fig. 9 – Available orders in Spartan
Fig. 10 – Battle status in Total War
Fig. 11 – Battle status in Spartan
Fig. 12 – Battle status in Legacy of Holy Castle
Fig. 13 – Medieval Battle depicted in (Bennett et al., 2005)
Fig. 14 – Contingent turning (DBM style on the left, Total War style on the right)
Fig. 15 – Tactics for a hero in Dragon Age: Origins (events on the left, actions on the right)
Fig. 16 – Comparing videogames’ flexibility when placing an army in the battlefield
Fig. 17 – Comparing videogames in terms of the shapes available for contingents
Fig. 18 – Comparing videogames in terms of the combat options available for contingents
Fig. 19 – Comparing videogames in terms of the options available for terrains
Fig. 20 – Comparing videogames in terms of the orders available for contingents
Fig. 21 – Comparing videogames in terms of a battle’s status
Fig. 22 – Comparing videogames in terms of contingent movement
Fig. 23 – Using different Contingent Categories in the same Sector Lines
Fig. 24 – Conceptual map
Fig. 25 – Roman formation in the Battle of Cannae
Fig. 26 – Carthaginian formation in the Battle of Cannae
Fig. 27 – Our platform architecture
Fig. 28 – The Iterative Design of a Graphical User Interface
Fig. 29 – Low-Fidelity Prototype #1
Fig. 30 – Low-Fidelity Prototype #2
Fig. 31 – Low-Fidelity Prototype #3
Fig. 32 – Image representing the task for Low-Fidelity Prototype #3
Fig. 33 – Functional Prototype #4
Fig. 34 – Functional Prototype #5
Fig. 35 – Event panel in the Functional Prototype #5
Fig. 36 – Assigning army contingents to slots
Fig. 37 – The battlefield as a 100 x 100 grid, with designated areas for contingents on both sides
Fig. 38 – Aligning the slots horizontally and vertically
Fig. 39 – A reactive agent with state
Fig. 40 – Agent state machine for the Battle Simulator
Fig. 41 – Battle Viewer’s panel showing information about each contingent
Fig. 42 – The Battle Viewer showing an ongoing battle
Fig. 43 – Category shapes for contingents (facing right)  
Fig. 44 – Battle Viewer panel showing information about a contingent  
Fig. 45 – Contingents performing various actions in the Battle Viewer  
Fig. 46 – The event panel showing an event being triggered during a battle  
Fig. 47 – The battle balance panel  
Fig. 48 – Subjects Game Experience and Battle Knowledge  
Fig. 49 – Formations for both armies in the playtest’s battle  
Fig. 50 – Results showing how well the three groups interpreted the playtest’s first battle  
Fig. 51 – Results showing subjects’ usage of the available features in the Tactical Planner  
Fig. 52 – Results showing subjects’ appreciation of the second battle

Tables

Tab. 1 – Available orders for the simulation and the tactical planning  
Tab. 2 – Functional Prototype #4 score for organization and simplicity  
Tab. 3 – Functional Prototype #5 score for organization and simplicity  
Tab. 4 – Weighted table for placing any army contingent in any slot  
Tab. 5 – Number of blocks required for a slot with four assigned contingents  
Tab. 6 – Factors and weights for determining an agent’s new target  
Tab. 7 – Calculating a new target for the “blue” agent  
Tab. 8 – Armies present in the playtest’s battle  
Tab. 9 – Results showing subjects’ interaction with the Battle Viewer  
Tab. 10 – Time spent on each application during the playtest

Acronyms

GUI – Graphical User Interface  
DBM – De Bellis Multitudinis (board game)  
MVC – Model-View-Controller (architecture)  
XML – Extensible Markup Language  
PHP – Hypertext Preprocessor  
LFP – Low-Fidelity Prototype  
FP – Functional Prototype
Definitions

Because most of the readers are not familiar with the terms used in this work, we added this section to list all the definitions that we find important to understand the remaining chapters, especially when it comes to terms commonly used in ancient or medieval battles, which are at the core of our work. Some terms that might be missing in this section can be found in the Conceptual Model, in Section 3.1; others may be repeated there for context purposes.

In order for an ancient or medieval battle to take place, it required three essential elements: two opposing armies (groups of soldiers that fight for the same side), two opposing generals (one leading each army) and a battlefield. In some battles there was more than one army fighting for the same side, each one with its own general, and those generals would agree on a plan of attack (i.e. a tactical plan) prior to the battle taking place.

Soldiers were organized in contingents, groups of soldiers that trained together, using the same type of armor and weapons. Soldiers either trained for hand-to-hand (or close range) combat, which required a melee weapon such as a sword, spear, axe or mace, or they trained for ranged combat, which required a ranged weapon such as a bow, crossbow or javelin (a short spear that is thrown at the enemy). In some cases, soldiers were trained to use artillery weapons such as a cannon, catapult or trebuchet.

The wealthiest soldiers -- usually nobles and high-ranking knights\(^1\) -- had a horse and trained for combat on it: they formed the cavalry of the army. The remaining soldiers moved on foot and formed the infantry. Most of the weapons that a soldier could use while on foot could also be used while on a horse.

On the battlefield, generals usually placed their soldiers in long straight lines, arranging each contingent in a rectangular shape (preferring width over depth) -- that was the most common formation. Some generals used different shapes for their contingents, like placing the cavalry in a wedge formation (looking like a triangle with the tip being in the front of the contingent) to thrust into the enemy line, try to breach it and divide it in two.

Formations on the battlefield usually had three sectors, or areas: the center, where most part of the army was placed, and the left and right flanks, usually used for cavalry. The flanks, placed on both ends (left and right) of the formation, were used to attack the enemy from its side. In some battles, like the Battle of Cannae described in section 1.2, the general could order his cavalry to attack the enemy from its rear, while the enemy was busy from the front. Attacking from the rear could be even more devastating than attacking from the flank.

Apart from the common orders referred in the previous paragraph -- frontal attack, flank attack and rear attack -- which were usually associated with melee contingents, ranged contingents could be assigned a different order like skirmishing, which consisted in keeping a safe distance from enemy soldiers and firing arrows at them whenever possible. This safety

\(^1\) [http://en.wikipedia.org/wiki/Horses_in_the_Middle_Ages](http://en.wikipedia.org/wiki/Horses_in_the_Middle_Ages)
distance was necessary, because archers were important in an army, and they were usually not very skilled with melee weapons, thus trying to avoid hand-to-hand combat.

Battles usually had five stages, or phases: the **start of the battle**, when each side had its troops placed in the battlefield and a considerable space between them called “No man’s land”. When both sides started walking towards each other, the **ranged phase** began. At this point, archers and other ranged soldiers were in firing distance of their enemy and started shooting arrows. After that, the **charge phase** took place, the stage when both sides were close and started running (or charging) towards each other. The charge would end in a violent collision, killing many of the soldiers in the front lines. After the charge, the **melee phase** ensued, when soldiers started hand-to-hand combat. Eventually one side would break (i.e. lose its morale and will to fight) and **retreat**, making the other side start its **pursuit** to kill whatever remaining enemy soldiers they could reach. (Keegan, 1976)

From this point on, when we refer to a formation, the orders given to the sectors in that formation and the events they react to, as a whole, we will call it a **Tactical Plan**, or simply **Tactics**, for the sake of simplicity.
1 Introduction

In videogames based on the Ancient or Medieval ages, players usually play the role of a king responsible for his kingdom, whose usual tasks are the planting of crops to feed his people, the extraction of minerals to make weapons and armor, the chopping of trees to use the wood for houses and weapons, the extraction of stone from the ground to build fortresses and other buildings, the forging of alliances with some kingdoms, the starting of wars with others, and the recruiting of armies to protect their lands.

The battles are an important aspect in these videogames. A battle is a confrontation between two opposing sides, where each side is composed of one or more armies of soldiers. The battles we refer throughout this thesis are the battles that occurred before the invention of the machine gun, which drastically changed the way battles unfolded. We will be referring to the battles where big groups of soldiers (or contingents) advanced towards each other, disposed in one or more lines, and usually fought in hand-to-hand combat using weapons such as swords, axes and spears.

Various games have been made where they try to reproduce battles, be it videogames, board games or table-top games. We can divide these games into three categories: games where players can control their contingents during the battle by moving them around and issuing orders (e.g. Total War, Great Battles of Alexander, De Bellis Multitudinis\(^2\)), games where players can choose where to place their contingents and issue one order to each before the battle starts, having few or no choices during the actual battle (e.g. Spartan, Centurion, Legacy of Holy Castle), and games where players can only decide how many -- and what types of -- soldiers they have in their armies, having no control over battles (i.e. no control over where their soldiers are placed and what orders they are to follow), because all battles are processed automatically by the game (e.g. Almansur, Travian).

1.1 Problem

In videogames like Almansur and Travian, players have no control over the course of a battle, except recruit more soldiers into their armies or choose the best territory for them to fight. Either the battle results are calculated using a formula based on the soldiers on each side, or every battle progresses in much the same way, regardless of the armies involved. Real-time videogames like Total War, on the other hand, become repetitive when players want to use the same tactical plan in various battles, because they must do everything themselves. Players can control their contingents at any time, changing their orders as they please, something that was not realistic in ancient and medieval battles, due to communication issues.

\(^2\) [http://www.byzant.demon.co.uk/dbm.htm](http://www.byzant.demon.co.uk/dbm.htm)
To address these concerns, we intend to create a platform where players can prepare rich and varied tactics, prior to a battle taking place, similar to what generals did in ancient and medieval times. This way, we give players a middle ground between having no control (like in Almansur and Travian) and having too much control (like in Total War). Note that only allowing pre-planning is not a loss in realism. Before the radio was invented, battlefield communications were highly unstable, being based on couriers, which meant the general had to decide his tactical plan in advance in order to be successful. Every tactical plan prepared in our platform will be independent from any army, so players can reuse their tactics in various battles, regardless of the armies they have. This will avoid unnecessary repetitiveness.

Generals used different tactics to command their armies into battle, so this platform must allow for different orders to be given to each sector, and in different moments. Some generals would send the whole army into battle, while others would attack in waves, sending one line of contingents at a time. And other generals would attack the enemy from the flanks, in order to surround and annihilate them. The next section describes a well-known ancient battle between the Romans and the Carthaginians to show a concrete example of some of the tactics used in those times. It should be possible to recreate all -- or most of -- the ancient and medieval battles in our platform.

Aside from letting players create their own tactical plans, the platform must allow players to watch the battles in which their armies participated. Players must be able to watch their tactics in action, learn what tactics their enemies use, understand what is happening during the battle and why their tactics were a success or a failure, learn from their mistakes, and be able to improve their tactics according to what they have learned.

1.2 The Battle of Cannae

The Battle of Cannae is one of many examples we can use as motivation for our work. We chose this battle for its complexity in tactics and we described it in the Introduction in order to help the reader understand the level of complex decisions we want to achieve in our work.

The Battle of Cannae took place on August 2, 216 BC, near the town of Cannae in Southeast Italy. On one side was a Roman army composed of 80000 infantry and 6400 cavalry, while on the other side was a Carthaginian army composed of 40000 heavy infantry, 6000 light infantry and 10000 cavalry. (Healy, 1994)

The Roman general chose to place his infantry in the center of the formation and the cavalry split into the left and right flanks, with more density in the left one. He also chose extra depth instead of width, in order to break through the enemy’s center and win the battle. This was the usual Roman tactical plan, to breach the enemy’s center, dividing its army in two.

The Carthaginian general, who knew about Roman tactics, placed his least experienced infantry in the center and his cavalry divided in both flanks, with more density in the left one. The most experienced infantry was placed behind the cavalry, invisible to the
Roman general, and its purpose was to attack the flanks of the Roman’s center after their charge. The purpose of the Carthaginian cavalry was to defeat the Roman cavalry (which was weaker and outnumbered) and then attack the Roman’s center from the rear.

As we can see from Fig. 1, the Roman’s center advanced towards the Carthaginian’s center, and the Carthaginian cavalry advanced towards the Roman cavalry. While the Roman cavalry was being slaughtered, the Carthaginian center started to turn into a semicircle, by way of a controlled retreat, with the objective of luring the Roman center into a trap and giving the Carthaginian cavalry more time. The Romans were so eager to destroy their enemy’s center that they did not notice the strongest Carthaginians in their flanks (see Fig. 2). They started forming a wedge to drive deeper into the semicircle and, at the decisive moment, the Carthaginian general ordered his infantry to turn inwards and advance against the Roman flanks, creating what is called the pincer movement. Eventually the Carthaginian cavalry attacked the Roman infantry from the rear (see Fig. 2), the Roman’s advance came to a halt, realizing that they were trapped from all sides, and most of them were slaughtered where they stood. In the end, 66000 Roman infantry were killed and only 8000 men were lost in the Carthaginian army. Only 14000 Roman infantry and 370 Roman cavalry survived the battle.

Fig. 1 – Initial formations and orders in the Battle of Cannae

Figures 1 and 2 are sliced in two because they were printed in adjacent pages of (Healy, 1994)
1.3 Objectives

The main objective in this work is to allow players to design their tactical plans with ease. Players should be allowed to place various types of soldiers in their formation, wherever they want, and give one or more orders to each sector in the formation, where each of those orders will have a battle event associated to it and will only be executed if that event occurs.

In order to accomplish our main objective, we need to create a Tactical Planner in the form of a Graphical User Interface (or GUI), which will allow players to create new tactical plans, save them and load existing ones. We will be prototyping the Tactical Planner until we can conclude that our test subjects understand what they are asked to do with the GUI.

The second objective is to show battles to players, where each side has one or more available armies and a tactical plan to follow. Players should understand what is happening during a battle, what actions and decisions are taking place, so that they can assess if their tactical plan was successful or if it must be improved. In the end, there should be a battle report showing which side was the winner and the casualties for both sides.

In order to accomplish our second objective, we will create a Battle Simulator and a Battle Viewer in 2D. The Battle Simulator will be a MultiAgent System, where each contingent is controlled by an Agent and the battlefield will be the Environment in which those agents exist. The Battle Viewer will show players the battles they participated in, viewed from the top. Each side will be assigned a different color (red for one and blue for the other, which is usual in books that depict ancient and medieval battles), and each contingent will be shown as a rectangle with its respective color and a representation of its soldier type on top of it.
1.4 Dissertation Layout

The rest of this paper will be divided into 6 chapters.

In Chapter 2 we analyze a wide variety of videogames (and a table-top game) that include ancient and medieval battles, regarding what options are offered to players before a battle takes place and, in some cases, during that battle. Some books on medieval battles are also analyzed, regarding how they represent battles in one or more images. We also compare these videogames and books according to various aspects like troop placing, available orders or how battles are shown to players, and we mention which ideas seem best for our work, according to our objectives.

In Chapter 3 we introduce the Conceptual Model that supports all our work, so the reader can understand the various elements involved in the platform. We also present the platform’s architecture, showing how the Tactical Planner, the Battle Simulator and the Battle Viewer connect to each other.

In Chapter 4 we describe the Tactical Planner, which will allow players to create their tactical plans to be used in later battles. We also list the various prototypes created and the evolution they had from the previous ones, along with the results from tests done to those prototypes.

In Chapter 5 we describe the Battle Simulator as a MultiAgent System, we show how the contingents in an army are placed in the battlefield, according to the formation in a tactical plan, we show how the agents in the Battle Simulator work, we describe the Battle Viewer, which presents each battle to players and helps them understand what is happening during the battle, like showing the battle’s events, which side is winning or where the contingents are.

In Chapter 6, we analyze the results of a playtest done to the entire platform, using three groups of subjects: five subjects inexperienced in ancient and medieval tactics who had no introduction to the platform, five experienced subjects who had no introduction to the platform, and five inexperienced subjects who were introduced to the platform before beginning the playtest.

Finally, in Chapter 7 we take some conclusions about our work and list the ideas we have for future work on this platform.
2 State of the Art

In this chapter we will start by analyzing six videogames and one table-top miniature war game that handle ancient and medieval battles differently, in terms of what the player is allowed to do before and during these battles. The way those games handle battles and the options players have will be compared, according to our objectives. We will sometimes refer books that depict historical battles and relate them to the way we chose to show battles in the Battle Viewer.

The videogames that will be analyzed in this chapter are the following:

1. **Medieval II - Total War**, where battles are shown in real time and players have the possibility to change tactics during the action. Players can also place their contingents in the battlefield before the battle starts.

2. **Spartan - Gates of Troy**, where players can place their contingents in the battlefield, before the battle starts, and give each contingent one order to follow during the battle. The battle is shown in real time, and while it is going on players can only order their troops to charge, rally or retreat.

3. **Great Battles of Alexander**, where battles are handled in turns, giving players the option to change tactics during their turn. Players are not allowed to place troops on the battlefield, since this game reproduces historical battles.

4. **Centurion - Defender of Rome**, where players can only choose a formation and a tactical move for their entire army. Battles are then shown in real time and players can pause it at any time. During the action, players can see the status and the order given to any contingent, and change the contingent’s order by telling it to either move straight towards the enemy or retreat.

5. **Legacy of Holy Castle**, where players define offensive and defensive tactical plans in a grid, prior to a battle taking place, and when a battle occurs the respective plan is used to place any available soldiers in formation. Every group of soldiers can only attack or defend. Battles are shown in turns after being simulated.

6. **Dragon Age: Origins**, where players control a group of heroes that are fighting their enemies, and can make tactical choices for each hero before (or during) combat, in order for the whole group to be successful in combat. These tactical choices consist of each hero reacting to certain events during each combat and performing a specific action upon it.

The table-top miniature war game that will be analyzed is **De Bellis Multitudinis**. The aim of DBM is to simulate large battles where the armies at both sides are chosen from published Army Lists, using a points system to select roughly equal (i.e. balanced) armies. In DBM, one of the players chooses the design of the battlefield, then both players place their troops on the battlefield and start moving their contingents to battle.
Sections 2.1 to 2.3 focus on what players can do in order to prepare their armies for battle, which will be useful for the Tactical Planner (see Chapter 4). Sections 2.4 and 2.5 focus on what players see during the course of battle, which will be useful for the Battle Viewer (see Section 5.5). Section 2.6 refers the event-driven behavior present in Dragon Age, which we want to use for our agents. Section 2.7 compares the various games and books referred throughout Sections 2.1 to 2.5, according to the objectives for this work.

2.1 Placing an army on the battlefield

One of the important aspects in ancient or medieval battles was the placing of contingents in the battlefield. Depending on the enemy's forces, a general would try to adapt his current army to ensure the best outcome in battle. For instance, putting archers in the front line inflicted more casualties on the enemy, because the shots were aimed straight at the soldiers. If the archers were placed in the second line, they had to fire their arrows over the first line, which inflicted less casualties on the enemy and increased the chance to miss. Thus, it is important that videogames allow players to place their troops on the battlefield as they see fit, allowing for different tactics and outcomes.

Medieval II - Total War

At the start of each battle, players are given an area inside the battlefield where they can place their contingents. Although the contingents are already placed in a tactical formation when the battlefield is loaded (see Fig. 3), it might not always be the best one and there is no restriction as to the changes players can make. And since the battlefield works with vectors instead of a grid, players can place their contingents wherever they want, turning them wherever they like, as long as they do not leave the designated area.

![Fig. 3 – Placing the troops in a Total War battlefield](image-url)
Spartan - Gates of Troy

In each battle, players are given an area inside the battlefield where they can place their contingents (see Fig. 4). Contingents are usually placed randomly, which forces players to move all of them into a tactical position. Unlike Total War, Spartan uses a grid to place the contingents and all contingents are facing the enemy when the battle starts, which somewhat limits the player's options. Also, contingents cannot overlap (i.e. cannot be placed over one another), unlike Total War.

![Fig. 4 – Placing the troops in a Spartan battlefield](image)

Great Battles of Alexander

Alexander does not allow players to place an army on the battlefield, since its purpose is to reproduce historical battles fought by Alexander the Great. When players choose one of the historical battles, they realize that their army is already in formation and the battle starts right away (see Fig. 5). One can observe, by playing the various battles, that the formations are usually wide and have a small depth (two lines at the most). That does not hold true for the other games analyzed, where players can make use of three or more lines.
When a battle takes place, players can only select one of four formations for the entire army: a balanced formation, a wedge formation, a “strong right” formation or a “strong left” formation. The rest is up to the game, which tries to place the players’ current armies as best as it can, according to the chosen formation (see Fig. 6). In the balanced formation, contingents are placed evenly across the lines in the formation. In the wedge formation, most of the contingents are placed in the center, and few contingents are placed in the flanks. In the “strong right” formation, most of the contingents are placed on the right side of the formation. In the “strong left” formation, most of the contingents are placed on the left side of the formation.

Fig. 5 – Troops already placed in an Alexander battlefield

Centurion - Defender of Rome

When a battle takes place, players can only select one of four formations for the entire army: a balanced formation, a wedge formation, a “strong right” formation or a “strong left” formation. The rest is up to the game, which tries to place the players’ current armies as best as it can, according to the chosen formation (see Fig. 6). In the balanced formation, contingents are placed evenly across the lines in the formation. In the wedge formation, most of the contingents are placed in the center, and few contingents are placed in the flanks. In the “strong right” formation, most of the contingents are placed on the right side of the formation. In the “strong left” formation, most of the contingents are placed on the left side of the formation.
Legacy of Holy Castle

Unlike the other videogames we analyzed, in Legacy players do not control their armies before the start of a battle; instead, they create tactical plans independently and then decide which ones to use for future battles. These plans can be either defensive or offensive: defensive plans are used when players’ villages are under attack, and offensive plans are used when players attack enemies’ villages. This was an idea that we found interesting to use for the Tactical Planner, having the tactical plans independent of any specific army or battle.

The battlefield in Legacy is made of a grid (see Fig. 7) and each player has half the battlefield to place his troops. Since players do not have a concrete army to place in the battlefield, they have to decide what types of soldiers they want and where to place them in the formation. Since the battlefield is a grid, and much smaller than Spartan’s, this gives players less flexibility when placing their armies in the battlefield. As in Spartan, all groups of soldiers are facing the enemy.
De Bellis Multitudinis

In DBM, one of the players is assigned to design the battlefield by placing a river, some trees, hills, and other terrain features. After that both players place their contingents, which are grouped into four Commands, in the battlefield: player A places command A1, then player B places command B1, then player A places command A2, and so on. Commands can be placed freely in the battlefield (there is no grid involved since this is a table-top game) as long as players leave an empty area between both armies called “No Man’s Land” (see Fig. 8).
2.2 Shapes, combat options and terrain

Shapes and combat options given to contingents were an important aspect in ancient and medieval battles, because they influenced the outcome of a battle. For instance, making a cavalry charge in a wedge formation was usually much more effective than doing it with a rectangular shape. Archers, specially longbowmen, were fundamental in an army, and as such they needed to be kept at a safe distance from the enemy. Telling them to skirmish (shoot arrows while avoiding a melee confrontation) was a way to keep them alive. When facing archers, having soldiers in a loose formation (keeping spaces between them) resulted in lower casualties.

Even the choice of terrain was important, because each unit reacted differently to the various types of terrain: cavalry had a hard time moving in rough terrain, preferring flat and open spaces, while archers preferred higher ground for longer range.

Medieval II - Total War

Total War gives players a high number of combat options for their contingents, which comes as no surprise given players can change their tactics during the battle. All contingents have the ability to be in a loose formation, reducing enemy ranged damage. Archers can skirmish, avoiding melee confrontation, which is their weakness. Archers can also use flaming arrows in some cases, which has lower accuracy but causes panic in the enemy’s lines. Players can also choose between a contingent running or walking.

When it comes to terrain, it isn’t clear if grassy terrain or sandy terrain makes any difference in a contingent’s outcome, but forests are very suitable if units want to avoid enemy
fire or cavalry charges. Cavalry charges can also be stopped if any obstacle is on their way, like a house, a wall or a cluster of rocks. Bodies of water cannot be crossed, so they are suitable for defense purposes. Elevation is not very common in Total War, but players will notice a speed reduction when going up a hill with cavalry, which will also reduce the damage inflicted by their charge.

Shapes, on the other hand, are not that common. The rectangular shape is the only one available, although players can resize that rectangle as they see fit (increasing or decreasing its width, which respectively decreases or increases its depth). Cavalry can be placed in a wedge shape for increased damage in a charge.

**Spartan - Gates of Troy**

Spartan comes with varied shapes for the contingents (see Fig. 4): players can shape contingents into three different rectangles (varying in width and depth), into a wedge (to break an enemy line), into a V-shape (to trap the enemy within) or into a diagonal line. Inexperienced contingents only have the rectangular shapes available.

When it comes to the combat options players can choose for a contingent, they can only order ranged units not to shoot any arrows and stick to melee combat (although the game considers that an order). Loose or tight formations are not an option; instead, depending on the type of contingent -- melee units or ranged units -- the available shapes will either be tight or loose, respectively.

Terrain is a big factor in Spartan's battles. When players are on the tactical screen, they can see where the battle is going to take place and the difference in terrain types is evident (see Fig. 4). Depending on the contingents a player has (and his enemy's contingents as well), choosing the right terrain for each contingent to fight may turn the tide of the battle. There are various types of terrain: clear ground, areas with bushes or rocks, sandy terrain, and even forests. There are no bodies of water in these battlefields, nor elevation.

**Great Battles of Alexander**

Alexander has no available shapes or combat options for contingents. What sets Alexander apart from the other games is the use of different types of terrain and elevation: players can choose to move their units to higher ground, giving them an advantage when in combat, or they can use forests to cover their units from enemy fire. Rivers can be crossed, although some men might die during the cross, and the player can try to lure the enemy into crossing a river.

**Centurion, Legacy and DBM**

Centurion and Legacy give players no combat options or shapes for contingents except the ones already set when the battle begins. All contingents are placed in a rectangular shape and there is no possible choice between loose or tight formations. Even terrains are homogeneous in each battle.
DBM does not allow players to change the shape of a contingent, for these are already rectangular and with a pre-defined size (see Fig. 8). The only thing players can do is move two contingents side by side, making it look like it is a larger rectangle. There is also no option for contingents to switch between tight and loose formations. Concerning the battlefield terrain, DBM is very rich, allowing one player to put in a river, some trees, sandy areas, swamps, hills, among other things.

2.3 Available orders for the army

Another important aspect in ancient and medieval battles were the orders given to contingents: who would strike first, where they would strike, or when to strike. That made the difference between a good general and a bad general.

Medieval II - Total War

While Total War is a videogame where players control each battle in real time, it has a short variety of basic orders available, like moving a contingent to a certain area in the battlefield or attacking a specific enemy contingent. Controlling all contingents in real time allows players to reproduce complex battle tactics from historical battles, because they can choose when each contingent will attack and from where.

Concerning movement, players can order a contingent to either walk or run to a certain area (or towards a certain target) in the battlefield. Running contingents will get tired more quickly, decreasing their combat performance.

Concerning combat, players can order their contingents to either attack or charge an enemy contingent. The first choice orders soldiers to walk towards the enemy in a straight line, and when in range they run towards them for the first charge. The second choice orders them to charge right away, which might tire them if the distance is too long and decrease their performance. Archers will start shooting arrows as soon as they are in range, when given the order to attack.

Spartan - Gates of Troy

Every contingent in Spartan has the same set of eight orders available (see Fig. 9) and can only be given one in each battle. These orders can be frontal attacks (with or without delay), frontal charges, flank attacks, outflank attacks (i.e. to attack the enemy from the rear), or two other orders specific for ranged units, where the first sends them towards the closest enemy contingent in sight, and the second orders them not to fire their arrows and rely only on their melee weapons.
During each battle, players have three orders that can be given only once and to the entire army: Charge, Rally and Retreat. Charge orders every soldier to run towards the nearest enemy. The Rally order slightly increases the contingents’ morale. Retreat orders every contingent to run away from the battlefield.

**Great Battles of Alexander**

Alexander is a videogame where players control each battle in turns. It has a short variety of orders available, just as Total War, but players can create complex tactics in the long run. Players can move contingents around as they please, as long as they have enough points left in that turn. The downside is that contingents can only walk and are restricted to the six orientations in a hexagon, since the battlefield is set on a hex grid.

There are only two orders related to combat: attack (when players click on an enemy contingent, making a contingent move towards the enemy and engage if in melee range) and shoot arrows (also when clicking an enemy contingent, making a contingent move towards the enemy and shoot a volley of arrows when in range).

**Centurion - Defender of Rome**

After choosing a formation for his army, the player can see the formation his enemy chose for his army and decide which order is best to defeat him. Orders (called Tactics in Centurion) are shown on a list after the player chooses his formation, and the player has to be familiar with them to understand what they do and what their pros and cons are. There are specific tactics for each type of formation available, and a few of these tactics were named after famous battles, like the Cannae Tactic, named after the Battle of Cannae (see Section 1.2).

During each battle, players can pause the game and check their army’s tactical plan, in this case, the movement each contingent is supposed to follow. When the game is paused, players are allowed to give one of two orders for each contingent: to attack the enemy, which sends them straight towards the enemy, disregarding the tactical plan as a whole, or to retreat from the battle.
Legacy of Holy Castle

In Legacy, each group placed on the battlefield can only attack or defend, which are orders players understand. The attack order sends the soldiers in a straight line, engaging the enemy in melee combat (or ranged combat, if possible) until the soldiers reach the end of the battlefield. In melee combat, soldiers only engage the enemy in front of them, they never turn sideways. The same is not true for ranged combat. The only exception when soldiers move and attack sideways in melee combat is when they reach the end of the battlefield, at which point they will turn towards the enemy hero’s tent (see Fig. 7).

De Bellis Multitidinis

DBM is a turn-based game, like Alexander, so players have few orders available. Most of the game is spent moving contingents around the battlefield, trying to reach the enemy in order to engage in combat. Once two enemy contingents collide, they engage in melee combat. Archers are also available in DBM and can shoot their arrows from a short distance.

2.4 Battle status

Knowing what was happening in a battle was crucial for generals. Sometimes tactics needed to be changed, if ever there was an opportunity for it; other times they had to know when to give the next order to some contingents in reserve.

Videogames sometimes exaggerate the type of information given to players, because in a real battle generals did not know exactly how many men were alive in a contingent at any given moment, if that contingent was winning or losing the conflict, if that contingent was near its breaking point (close to retreating), how many arrows the archers had left, and so on. Nonetheless, we will analyze the videogames we are researching in terms of the information they show players during a battle.

Medieval II - Total War

Every battle in Total War has a huge amount of information available to the player at any given moment (see Fig. 10). There is a minimap showing where each contingent is on the battlefield, as well as the battlefield layout (trees, rivers, hills). There is the player’s contingent list, showing how many soldiers each contingent has, if they are stopped, walking or running, if they are under enemy fire, shooting arrows or in melee combat, how many arrows are left, or if they are fleeing. Finally, there is the battlefield itself, which shows a part of the battle from a 3D point of view, where the player can get more detailed info on every contingent (e.g. morale, tiredness), see where each soldier is, their actions, and the fight itself.
Spartan – Gates of Troy

Spartan does not show as much information during a battle as Total War. It has a minimap as well, showing where each soldier is. It has a contingent list, showing how many soldiers each contingent has and what their current morale is. And finally there is the battlefield itself, shown from a 3D point of view, where players can partially see the battle taking place (see Fig. 11).

The problem with Spartan is players may have a hard time matching each soldier in the battlefield to its respective contingent in the contingent’s list. However, it does not make any difference, since players can do almost nothing during a battle.
Great Battles of Alexander

In Alexander, players are shown a partial view of the battlefield from a 2D point of view. There is no minimap or contingent’s list. The status of a contingent can be seen by placing the mouse cursor over it. A box will show the contingent’s name, the type and number of soldiers in it, their quality in combat (the higher, the better) and how many hits they have taken (the lower, the better).

Centurion – Defender of Rome

In Centurion, the player is only shown the battlefield during a battle. There is no minimap (since the whole battlefield is visible) and there is no contingent’s list. Players can pause the game at any moment, check the morale and number of soldiers of any contingent (friend or foe), and they can also check the tactical plan for their army (i.e. the movement orders assigned to each contingent).

Legacy of Holy Castle

Legacy shows little information during a battle (see Fig. 12). After a battle takes place, players are shown a view of the entire battlefield and they can replay that battle by turns. In each turn soldiers move one or two squares on the grid (if they were ordered to attack) or stay in their position (if they were ordered to defend). When a soldier attacks another, the damage
inflicted is displayed in text, to the right of the battlefield view. Legacy only shows how healthy each soldier still is, there is no morale or tiredness involved in this game.

Fig. 12 – Battle status in Legacy of Holy Castle

De Bellis Multitudinis

Because DBM is not a videogame, the status of any battle is really simple: players have a view of the entire battlefield at any given moment (it's all on the table), and contingents are either alive (in the battlefield) or dead (removed from the battlefield). Apart from that, both players must memorize whose turn it is, what moves that player has already made to each of his commands (groups of contingents), and they must both confirm every action made (e.g. moving distances).

Books on medieval battles

Battles depicted in books have almost nothing in common with battles shown in a videogame. The sense of progression in books must be shown by means of arrows (see Fig. 13) and sometimes using a sequence of images (much like Legacy shows every battle to players), while in videogames players usually see battles in real time.

Contingents in books are represented either as rectangles with a specific shape inside, to indicate different types of contingents, or as groups of soldiers, where each soldier in the picture represents around 10 soldiers in the real battle.

The battle status is usually described in short sentences, numbered to better understand the battle progression. Apart from the images contained in books, each battle is described thoroughly: how many soldiers were suspected to be on each side (usually the numbers given by chronicles are exaggerated), how the battle progressed, who won and, finally, the aftermath of that battle.
2.5 Movement on the battlefield

Movement is a big issue in videogames and one that requires a lot of attention: there are collisions to be handled, obstacles to be avoided, paths to be found to reach a certain goal, contingents that need to move together while keeping their shapes, and so on. Some videogames allow really complex movement for their contingents, while others keep it relatively simple and lack realism.

Medieval II – Total War

Total War is very realistic when it comes to movement, but it comes with its glitches. Contingents can move in any direction: they can walk, run, rotate, accelerate and decelerate,
while soldiers in a contingent keep to their positions so the contingent moves as a whole, and try to avoid obstacles as best they can. Collisions are handled nicely, but not without their problems. A whole cavalry charge will come to a halt if there is a single rock blocking a horse's path, because the game's priority is to keep the contingent together. A collision between enemy contingents results in melee combat, while between friends it depends on the speed: if one or both are walking, they'll cross each other as best they can, but if one of them is running, it will come to a halt and try to cross the other by walking.

**Spartan – Gates of Troy**

Soldier movement in Spartan is fairly realistic, but although it avoids a problem faced in Total War, it has its limitations. Unlike Total War, there is no notion of a contingent as a whole when a battle starts. Every soldier is independent. Also, the kind of free movement allowed in Total War is nowhere to be seen in Spartan, where soldiers only move in straight lines without rotating, accelerating or decelerating. Collision works by simply not allowing two soldiers to be in the same position on the grid. If they are enemies, they engage in combat. If they are friends, the one to get there last tries to find another way around to reach his enemy.

**Great Battles of Alexander**

Movement in Alexander works on a hex grid, so it is fairly simple. Contingents can turn to any of the six sides in a hexagon, they can move freely around the battlefield, but there is no acceleration or deceleration. Collisions are handled by simply not allowing a contingent to move into a hex already occupied by another contingent. If those contingents are foes, though, they engage in combat. In case two friendly contingents collide and one of them is retreating, they both inflict some damage to the other because of their collision.

**Centurion – Defender of Rome**

Movement in Centurion is fairly simple, which is not surprising considering it is an old videogame. It works on a grid and soldiers can move to the front, the sides or the rear. There is no rotation, acceleration or deceleration in this game. Collisions between foes end in melee combat, and when two friendly contingents collide one of them waits until it has room to continue its movement.

**Legacy of Holy Castle**

In Legacy, soldiers can only walk to the front or towards the center, when they reach the end of the battlefield, because movement is based on a grid. Every turn, each soldier walks one or two squares either to the front or to the side. This game has no movement animation whatsoever. If two enemies collide, they engage in melee combat. If two friends collide, the one on the rear waits until he has room to move forward.
De Bellis Multitudinis

Contingent movement in DBM is fairly realistic. Considering that contingents are represented as rectangles, they never lose their cohesion or shape, which is exaggerated, but acceptable considering it is a table-top game. Also, because it is a table-top game, it is not possible for contingents to overlap, so collisions work between friends and foes.

One thing that sets this game apart from the other videogames is the rotation of a contingent. In Total War, for instance, when the player orders a contingent to turn, the turning “axis” is situated in the contingent’s center; in DBM, the “axis” is situated either on the left end of the contingent (if it wants to turn left) or on the right end (if it wants to turn right). DBM’s way of rotating contingents is the real way contingents turned in a real battle (see Fig. 14).

![Contingent turning](image)

Fig. 14 – Contingent turning (DBM style on the left, Total War style on the right)

2.6 Event-driven behavior

None of the games we have analyzed thus far allow players to make the contingents in their armies react to certain events that happen during a battle. Total War, Alexander and DBM, which are either in real time or turn-based, let players react to the events and change the orders given to their contingents at any point during a battle. What we are aiming for, though, is to allow players to decide what events their contingents should react to -- by this we mean the Agents controlling the contingents -- and what orders they should follow after each of those events. This should be done before the battle actually happens because, in our platform, players cannot control their armies during the battle.

Bioware’s Dragon Age: Origins is a Fantasy Role-Playing game where the player has control over a team of heroes that must work together to defeat their enemies. Part of the game revolves around combat situations and the player can prepare some tactical choices for each of his heroes in order for the whole group to be successful. Each of these tactical choices consists of the hero reacting to a certain event during combat and performing a specific action upon it. For instance, the player can make the Healer perform a healing spell (Action) on a friend if his health gets low (Event), or he can make the Archer keep a safe
distance from his enemies (Action) if they get too close (Event), because he is better suited for ranged combat.

The way Dragon Age does this is by showing players a list of events and actions for each hero (see Fig. 15), which players can change whenever they want, even during combat. When each event is triggered, the hero executes his corresponding action. Dragon Age is a successful game, according to Alexander (2010) and Reilly (2010), and the concept of giving heroes an event-driven behavior seems simple enough to use. We intend to integrate this concept into the platform so that players can make their contingents follow a different order when a certain event occurs during the battle. This will give battles more realism, as it is similar to the way generals planned their tactics before a battle.

Fig. 15 – Tactics for a hero in Dragon Age: Origins (events on the left, actions on the right)

2.7 Comparing solutions

In this section, we compare the videogames analyzed so far (as well as the books, when possible) according to each aspect referred in sections 2.1 to 2.5, so that we may describe the solution for our work. When necessary, we make a brief heuristic analysis of the videogames.

Placing an army on the battlefield

According to our analysis, Total War and DBM are the games that give players the most flexibility when placing their armies in the battlefield, because there is no restriction as to
where contingents can be placed, or where they are facing, except that there has to be a space between both armies called “No Man’s Land”.

Although Spartan and Legacy give players a lot of flexibility as to where they can place their contingents, they are based on a 2D grid and force all contingents to face the same direction.

Alexander and Centurion place the armies automatically in the battlefield, the first based on a formation the player chooses from a list, and the second because it is trying to reproduce known battles from Alexander’s campaign, therefore the armies have fixed positions in the battlefield.

<table>
<thead>
<tr>
<th>Placing flexibility</th>
<th>Alexander</th>
<th>Centurion</th>
<th>Legacy</th>
<th>Spartan</th>
<th>DBM</th>
<th>Total War</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>2D Hex Grid</td>
<td>2D Grid</td>
<td>2D Grid</td>
<td>2D Grid</td>
<td>3D Space</td>
<td>3D Space</td>
</tr>
<tr>
<td>Orientations</td>
<td>Varied</td>
<td>Same</td>
<td>Same</td>
<td>Same</td>
<td>Varied</td>
<td>Varied</td>
</tr>
<tr>
<td>No Man’s Land</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Overlapping</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Smallest entity</td>
<td>Contingent</td>
<td>Contingent</td>
<td>Soldier</td>
<td>Soldier</td>
<td>Contingent</td>
<td>Contingent</td>
</tr>
</tbody>
</table>

Fig. 16 – Comparing videogames’ flexibility when placing an army in the battlefield

In most historical battles depicted in books like (Bennett et al., 2005), (Devries et al., 2006), (Healy, 1994) and (Cowan, 2007), we can see that contingents were organized in long, straight lines, all facing the same direction, although in some battles they were separated into commands and would attack the enemy from different points, thus facing different directions in the battlefield.

From most of the battles we saw depicted, we reached a general layout for all formations. This layout contains a central sector, where most of the contingents are placed, and two flanks, which can either be used to extend the central lines or to attack the enemy from its flanks or rear. Some generals would also keep contingents in reserve to use them later in battle, which can also be done using our layout, since players will be able to have multiple lines in each sector, and the rear lines may be used as reserves.

For our platform, we will restrict the placing to a 2D grid and all contingents will be facing the same direction when the battle starts. We also want the tactical planning to be independent from any army a player controls, which means that when a battle occurs the placing of the contingents in the battlefield will be automatic and according to the choices the player made in the tactical plan. Because the tactical plan is independent from any army, the platform must be able to adapt any given army to the choices in the tactical plan.

Legacy, for instance, although having independent tactical plans, is not flexible when it comes to placing soldiers in the battlefield. If a player chooses to have spearmen in a certain slot, Legacy’s battle simulator will only place spearmen there. If there are no spearmen in the army, that slot will remain empty.
We want the platform to first place each type of contingent in its rightful slot (melee contingents in melee slots, ranged contingents in ranged slots, and so on), and if after that there are still some contingents left to place, they will be placed in the next best possible slot (e.g. ranged contingents in melee slots, because they carry a melee weapon too).

**Shapes**

<table>
<thead>
<tr>
<th>Number of options</th>
<th>Alexander, Centurion, Legacy, DBM</th>
<th>Total War</th>
<th>Spartan</th>
</tr>
</thead>
<tbody>
<tr>
<td>No shapes available</td>
<td>Rectangular</td>
<td>Rectangular, Wedge</td>
<td></td>
</tr>
<tr>
<td>Wedge (cavalry)</td>
<td></td>
<td>Diagonals, V-Shape</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 17 – Comparing videogames in terms of the shapes available for contingents

After analyzing the games in section 2.2, we concluded that Spartan is definitely the videogame with the most shapes available for a contingent. In fact, it is the only videogame with a wide variety of shapes to use in our lines. However, while those shapes may come in handy for certain tactics, we do not feel there is the need to overburden the Tactical Planner by adding this feature.

**Combat Options**

<table>
<thead>
<tr>
<th>Number of options</th>
<th>Alexander, Centurion, Legacy, DBM</th>
<th>Spartan</th>
<th>Total War</th>
</tr>
</thead>
<tbody>
<tr>
<td>No options available</td>
<td>Skirmish, Hold your fire</td>
<td>Skirmish, Hold your fire</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Walking / Running, Flaming arrows</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 18 – Comparing videogames in terms of the combat options available for contingents

Total War is the videogame with the most combat options available for contingents, as analyzed in section 2.2. For this work, however, we will only be using the “Skirmish” combat option, and it will be considered an order -- soldiers will walk towards the enemy, start shooting arrows when the enemy is in range and skirmish (i.e. move back) when the enemy gets too close. In a real battle, soldiers would only run when they were close to the enemy, so the option to run or walk is unnecessary. Flaming arrows will not be available in our work, since they were not that common in battles.

**Terrain**

In section 2.2 we mentioned how terrain was an important factor in battles, especially in Alexander and Spartan. The different types of terrain are evident in DBM, Spartan and Alexander, as players can easily distinguish them on the battlefield. As for their effects in
battle, Alexander shows how being in higher ground is clearly advantageous in melee combat. Spartan, while having many types of terrain and referring what type of terrain each unit prefers, fails to show the terrain effects in battle. Both Alexander and Total War show the effects of being inside a forest.

<table>
<thead>
<tr>
<th>Types of terrain</th>
<th>Centurion, Legacy</th>
<th>Total War</th>
<th>Spartan</th>
<th>DBM</th>
<th>Alexander</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forests</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>High ground</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Various terrains</td>
<td>No</td>
<td>Unclear</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Obstacles</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Rivers</td>
<td>No</td>
<td>Not crossable</td>
<td>No</td>
<td>Not crossable</td>
<td>Crossable</td>
</tr>
</tbody>
</table>

Though the various types of terrain had a fundamental role in ancient and medieval battles, as can be read in books like (Devries et al., 2006) and (Keegan, 1976), this work will not focus on handling different types of terrain inside the battlefield, but it is something to be thought about in the future. For this first phase, terrain will be homogeneous throughout the battlefield.

Orders

<table>
<thead>
<tr>
<th>Number of available orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy</td>
</tr>
<tr>
<td>Attack</td>
</tr>
<tr>
<td>Defend</td>
</tr>
<tr>
<td>Centurion</td>
</tr>
<tr>
<td>Army tactics*</td>
</tr>
<tr>
<td>Alexander, T. War, DBM</td>
</tr>
<tr>
<td>Attack</td>
</tr>
<tr>
<td>Defend</td>
</tr>
<tr>
<td>Movement orders</td>
</tr>
<tr>
<td>Spartan</td>
</tr>
<tr>
<td>Attack</td>
</tr>
<tr>
<td>Flank</td>
</tr>
<tr>
<td>Surround</td>
</tr>
<tr>
<td>Delayed attacks</td>
</tr>
<tr>
<td>Archer-specific orders</td>
</tr>
</tbody>
</table>
* Based on historical battles

When it comes to the orders available for a contingent, it isn’t hard to define which of the games analyzed is best suited for our work. Total War, DBM and Alexander are out of question, since they work in real time and our work will focus on giving orders before a battle takes place, which means giving more complex orders than the ones available — “move here” or “attack that”. Legacy has two orders available — attack or defend — which are really basic, and we want to give players more flexibility.

Centurion’s orders are very complex, as we saw in section 2.3. Players choose a tactic for the whole army, which translates into giving movement orders to each contingent in
the battlefield. When contingents collide with an enemy, they engage in combat; if victorious, they keep following their initial order. This solution has two flaws: first, players must be familiar with the available tactics to understand what each of them does; second, it doesn’t allow experienced players to design their own tactics.

Spartan, on the other hand, has a wide variety of orders and allows players to give a specific order for each contingent in the battlefield. It has two flaws though: some of those orders are hard to understand at first, and the tactical view (see Fig. 4) does not display the order given to each contingent, forcing players to either remember which orders they gave, or select each contingent to check its current order.

**Battle status**

Understanding what is happening in a battle is very important to our work, not just the tactical planning. Total War is the game that shows the most information about a battle, and we want to try and create a solution as close to it as possible: each contingent will show which action it is performing at any moment by means of an icon shown over the respective contingent, and placing the mouse over the contingent will show additional information like tiredness, the number of soldiers in it, their speed and their cohesion.

<table>
<thead>
<tr>
<th>Amount of information</th>
<th>DBM</th>
<th>Legacy</th>
<th>Centurion</th>
<th>Alexander</th>
<th>Spartan</th>
<th>Total War</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contingent health</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Contingent status</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Contingent action</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Battlefield</td>
<td>Whole</td>
<td>Whole</td>
<td>Whole</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
</tr>
<tr>
<td>Minimap</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Fig. 21 – Comparing videogames in terms of a battle’s status

Because we will need to have a point of view close to the battlefield, so players can distinguish each contingent and what is happening, there will be a minimap similar to the one in Spartan, since it is easy to understand. Total War’s minimap is not a good example to follow, because each contingent is represented by a big arrow, and these arrows overlap most of the times when the contingents are close to each other. Instead, we will follow Spartan’s example and represent each contingent by a small square or dot.

**Movement**

Total War is definitely the best videogame when it comes to movement, because players want battles to be as realistic as possible: contingents walk or run, rotate, avoid collisions with friends, everything that is expected in a real battle. Still, it has some glitches concerning contingent cohesion, because in a real battle some obstacles would be easily avoided, while in Total War they can stop a whole contingent just so it remains together.
Movement in DBM is very similar to Total War’s, although it has one extra detail that no other videogame has, concerning contingent rotation (shown in Fig. 14). We want our contingents to behave like the ones in DBM, in terms of rotation.

<table>
<thead>
<tr>
<th>Movement realism</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Movement</strong></td>
</tr>
<tr>
<td>2D</td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
</tr>
<tr>
<td><strong>Collisions</strong></td>
</tr>
<tr>
<td><strong>Cohesion</strong></td>
</tr>
</tbody>
</table>

* Each contingent in Legacy is represented by one soldier, so there is no concept of cohesion

Fig. 22 – Comparing videogames in terms of contingent movement

Spartan does not have Total War’s problem because every soldier is independent once the battle starts, so it does not have to move close to its contingent. But movement in Spartan is not as natural, since it lacks accelerations and rotations, and that has a negative impact on realism. Still, the notion of soldiers being in a contingent is important, because they trained together and therefore trusted each other. A soldier will fight a lot harder if he is beside his friends, making cohesion an important aspect in a contingent.

The other three games (Alexander, Legacy and Centurion) are just too simple, or work differently from what we have in mind for our work. Our solution is to be somewhat a middle ground between Total War, Spartan and DBM. We will represent each contingent as a rectangle, like in DBM, and it will move in the battlefield without losing its shape. Movement will be fluid like in Total War and DBM, though we won’t complicate our simulator with accelerations, because battles are to be shown at a speed faster than normal -- it would take too long to watch a one hour battle, which was the average time for a short battle.
3 Conceptual Model

In this chapter, we will be focusing on the Conceptual Model for this work, in order to clarify what we want from the Tactical Planner (i.e. what we want players to be able to do as a general), the Battle Simulator and the Battle Viewer. For that, we need to understand what a battle consists of, so we can transmit those concepts to our players.

3.1 Concepts

In Section 1.2, we described the Battle of Cannae without going into detail about the various concepts present there. This section is intended to describe those concepts, as well as others that were not mentioned and need to be addressed, in order for our Conceptual Model to allow the platform to recreate other important battles in History. Some of the concepts in this section are mentioned by Keegan (1976, 2004).

General

The first concept we will introduce is the general, which is the role of the player. The general is the man who commands his army into battle, who places his soldiers in the battlefield and tells them what orders to follow. Some generals would fight together with their men, while others would stay behind, watching the battle unfold and signaling when new orders were to be executed, generally by using pre-planned signals like waving a flag or sounding a horn. In some of the games we analyzed in chapter 2, like Total War, Spartan and Alexander, generals would be in the battlefield with their troops and could be killed. For our work, players will be the general who is just watching the battle and who has planned it before the actual confrontation.

It is important to notice that giving new orders during a battle was hard, because messengers were very inefficient. Thus, the placing of the troops and the orders they were to follow would have to be planned before the actual battle took place. The pre-planned signals would be used to tell soldiers when to execute some of those orders, in case something happened during the battle that the general was expecting.

Army

An army is a collection of contingents fighting on the same side, usually lead by the same general. It was common in many historical battles for various armies to fight on the same side, each one lead by a different general. For our work, however, only one tactical plan will be used for all the armies on each side, which means that only one general will be in command.
Contingents and Contingent Categories

A contingent is a group of soldiers who perform similar roles: they carry the same weapon and armor, they may walk or ride a horse, and they train and fight together, which gives them a similar battle experience. For our work, we will divide contingents into five categories:

1. Melee Infantry, composed of soldiers who walk and wield a hand-to-hand weapon like a sword, spear, pike, axe or mace;
2. Ranged Infantry, composed of soldiers who walk and carry a long-range weapon like a shortbow, longbow, crossbow or javelin;
3. Melee Cavalry, composed of soldiers who ride a horse and wield a hand-to-hand weapon;
4. Ranged Cavalry, composed of soldiers who ride a horse and carry a long-range weapon;
5. Artillery, composed of heavy machinery like cannons, catapults or trebuchets.

It is important to notice that, while some soldiers might be focused on melee or ranged combat, they might be carrying another weapon. Archers would carry a sword in case the enemy reached their lines, although they were not very experienced with it.

Formation, Sector, Sector Line

A formation is the way a general organizes his contingents in the battlefield. It was common for a formation to have three sectors: the center, the left flank and the right flank, and contingents would be placed in one or more lines – the sector lines. According to the general’s tactics, archers could be placed in the front (or rear) lines, and cavalry could be placed in the flanks or in the center of the formation, among many other choices a general could make.

Formations in our work will have a fixed width but its depth will be variable, meaning players will be able to add lines to -- or remove lines from -- their formation. In any tactical plan, all the sectors in the formation always have the same number of lines, at any time, which means adding a line to the formation adds a line to all the sectors, and it works the same way when removing a line. Players are allowed to leave some of the sector lines empty, if they wish.

Slot

Each sector line is composed of various slots so that, for instance, players can place Melee Infantry besides Ranged Infantry. Each slot can only be assigned a contingent category, but each sector line can have multiple categories, hence the use of slots to enable that option. Figure 23 shows an example of different contingent categories in different slots.

Formations in the Tactical Planner will be independent from any army. This means that players will not be placing actual contingents in their formation, but instead they will decide where they want a certain category of soldiers to be placed (e.g. where they want their...
archers to be, where they want their cavalry to be, and so on, as shown in Fig. 23). This will allow players to use their tactical plans in different battles and with different armies. It will be the job of the Battle Simulator to understand the choices a player made in his tactical plan and associate the actual contingents in the player's army to the chosen categories in the formation, as best as possible.

![Ranged Inf. Melee Inf. Melee Inf. Ranged Inf.](image)

![Ranged Cav. Melee Inf. Ranged Inf. Ranged Cav.](image)

![Melee Cav. Melee Cav.](image)

Fig. 23 – Using different Contingent Categories in the same Sector Lines (the top line in this figure shows the front line in the formation)

**Rank**

Some generals placed their strongest soldiers in a specific area of the formation, for a specific purpose, and that can be done in our work by using ranks. In our model, that “area” will be one or more specific slots. Each slot will have a rank associated to it, and players will be allowed to change each of those ranks.

Three ranks will be available to players when designing their tactical plan: low (for the weakest soldiers), average, and high (for the strongest soldiers). Ranks are only meant to be used when the player wants a specific area in his formation to have the strongest (or weakest) soldiers. By default, every slot will be assigned the average rank.

**Orders and Available Orders**

Players will be able to associate one or more orders to each sector line in the formation, and those orders will be executed by all the contingents placed in that line during a battle. Some of these orders will have pre-conditions that must be met in order for contingents to execute them, otherwise they will keep doing what they were doing before. Also, contingents will stop executing their current order if a certain end-condition is met, or if an order with higher priority is triggered (we explain priorities later in this section). The following table lists and describes all the orders that will be available in our work, together with their pre-conditions and end-conditions.
<table>
<thead>
<tr>
<th>Order</th>
<th>Situational Pre-Condition</th>
<th>End-Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defend</td>
<td>Not in melee combat; Not in ranged combat</td>
<td>Enters melee combat; Enters ranged combat</td>
<td>The contingent stays where it is and only engages in melee combat if the enemy attacks it, or in ranged combat if an enemy contingent is inside the shooting range.</td>
</tr>
<tr>
<td>Ranged attack</td>
<td>Not in melee combat; Enemy group in sight; Only ranged units placed in sector line</td>
<td>Enters melee combat; No enemy in sight</td>
<td>The contingent advances until enemy is in range, then shoots arrows.</td>
</tr>
<tr>
<td>Skirmish</td>
<td>Not in melee combat; Only ranged units placed in sector line</td>
<td>Enters melee combat; No enemy in sight</td>
<td>The contingent advances towards the enemy, then shoots when enemy is in range or retreats and shoots if the enemy is closer than half the firing range.</td>
</tr>
<tr>
<td>Melee attack</td>
<td></td>
<td>Enemy retreats</td>
<td>The contingent advances straight ahead and, if after some time there is no contact, it turns towards the nearest enemy in sight and advances towards it.</td>
</tr>
<tr>
<td>Flank attack</td>
<td>Not in melee combat; Enemy group in sight</td>
<td>Enters melee combat; No enemy in sight</td>
<td>The contingent advances, avoiding the enemy until its flank or rear is exposed, then advances towards the enemy and engages in melee combat.</td>
</tr>
<tr>
<td>Controlled retreat</td>
<td></td>
<td>No enemy in sight</td>
<td>The contingent retreats slowly, while keeping the enemy engaged in melee combat.</td>
</tr>
<tr>
<td>Pursuit</td>
<td>Not in melee combat; Enemy is retreating</td>
<td>No retreating enemy in sight</td>
<td>The contingent advances towards a fleeing enemy, while engaging in melee or ranged combat when in range.</td>
</tr>
<tr>
<td>Rear attack</td>
<td>Not in melee combat</td>
<td></td>
<td>The contingent advances, avoiding the enemy until its rear is exposed, then advances towards the enemy and engages in melee combat.</td>
</tr>
<tr>
<td>Retreat</td>
<td></td>
<td>Out of battlefield</td>
<td>The contingent disengages the enemy and retreats as fast as possible from the battlefield.</td>
</tr>
<tr>
<td>Fake retreat</td>
<td>Target of attack</td>
<td>No enemy in sight</td>
<td>The contingent runs away from the enemy (as if it were retreating) for a short time, then turns back towards the enemy.</td>
</tr>
</tbody>
</table>

Tab. 1 – Available orders for the simulation and the tactical planning

Each contingent may know how to execute all (or only some) of the orders on this table. When the player sets which contingents he wants in a sector line, the only orders available for that line will be the ones that all contingents on that line know how to execute.
For instance, if a certain sector line is meant to have Melee Infantry and Ranged Infantry, the orders Ranged Attack and Skirmish will not be available to that sector line, since Melee Infantry contingents do not know how to execute those orders, because they do not carry ranged weapons.

Event

Each order given to a sector line will have an event associated to it. Players will be able to give different orders to the same sector line and each one will be associated to a different event that may occur during a battle. Events are divided into two categories: global events, which trigger when that event happens anywhere in the battlefield, and local events, which trigger when that event happens to a specific sector in the battlefield, friend or foe. We want to have both global and local events, because we want sectors to be able to react to events that concern the whole army, or just a part of it. Events are our way of allowing players to use the pre-planned signals we mentioned at the beginning of this section.

There are five global events in our work, relative to the whole battle:
1. Battle : Start, which triggers as soon as the battle starts
2. Battle : Ranged Phase, which triggers as soon as one of the contingents (on any side) starts shooting arrows at an enemy contingent
3. Battle : Melee Phase, which triggers as soon as one of the contingents (on any side) shocks against an enemy contingent, engaging in melee combat after that shock
4. Battle : Enemy Retreat, which happens when the first enemy contingent retreats from the battle
5. Battle : Friendly Retreat, which happens when the first friendly contingent retreats from the battle

The local events, which are relative to specific sectors on any side, may be more complex to use by inexperienced players and harder to predict if they will ever happen in a battle. The local events are as follows:
1. Sector : Target of : Ranged attack, which triggers if a specific sector is the target of a ranged attack by its enemy
2. Sector : Target of : Melee attack, which triggers if a specific sector is the target of a melee attack by its enemy
3. Sector : Target of : Flank attack, which triggers if a specific sector is the target of a flank attack by its enemy
4. Sector : Retreats, which happens when the specified sector retreats from the battle

The Sector referred in the local events may be one of the following:
1. Friendly : This, which refers to the sector that is being assigned the event
2. Friendly : Center, which refers to the center of the player’s formation
3. Friendly : Flank, which refers to any of the two flanks of the player’s formation
4. Friendly : Left flank, which refers to the left flank of the player’s formation
5. Friendly : Right flank, which refers to the right flank of the player’s formation
6. Enemy : Opposing, which refers to the enemy sector in front of the sector that is being assigned the event
7. Enemy : Center, which refers to the center of the enemy’s formation
8. Enemy : Flank, which refers to any of the two flanks of the enemy’s formation
9. Enemy : Left flank, which refers to the left flank of the enemy’s formation
10. Enemy : Right flank, which refers to the right flank of the enemy’s formation

Rules and Priorities

As we referred, players will be able to give various orders to the same sector line and each will have an event associated to it. We will call a Rule to each of these orders that have an event associated to them. There may be some conflicts if two different orders are to be executed at the same time, because two different events happened close to each other. The way we deal with this problem is by giving a priority to each rule. Players will be able to set one or more rules for each sector line, and their priority on the list will matter, because it will define which gets executed.

From time to time, the rules in each sector line will be checked to see if their condition was met (i.e. if their event was triggered). Then that sector line will follow the order of the highest priority rule that met its condition. Consider the following example:

<table>
<thead>
<tr>
<th>Priority</th>
<th>Condition / Event</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Battle : Start</td>
<td>Ranged attack</td>
</tr>
<tr>
<td>2</td>
<td>Friendly : This : Target of : Melee attack</td>
<td>Controlled retreat</td>
</tr>
<tr>
<td>3</td>
<td>Enemy : Opposing : Target of : Flank attack</td>
<td>Melee attack</td>
</tr>
<tr>
<td>4</td>
<td>Friendly : Flank : Retreats</td>
<td>Retreat</td>
</tr>
</tbody>
</table>

According to these rules, the corresponding sector line will start the battle by engaging the enemy in ranged combat (Rule #1), because the events from the other three rules have not been triggered yet, since both armies have some space between them when the battle starts. When the enemy reaches this sector line and starts melee combat, this triggers Rule #2 (with a higher priority than Rule #1), which sends this sector line into a controlled retreat. If the enemy is attacked from one of its flanks, Rule #3 (with a higher priority than Rules #2 and #1) will be executed, and this sector line will engage the enemy in melee combat. The last rule orders this sector line to retreat if any of the flanks on the formation start retreating, to try and save the soldiers on this sector line, so this rule should have the highest priority of all.
Event Persistence

One aspect that needs to be addressed in the conceptual model is the persistence of global and local events throughout the battle. Rules that have a global event as a condition will always be listed for execution once that event is triggered; the same is not true for local events. Although rules with global events will always be listed once those events are triggered, priorities still apply: only the rule with the highest priority will be executed by its corresponding sector line.

For instance, using the previous example, Rule #1 will always be listed because, after the battle starts, it is always true that the battle has started. Now consider that the sector line becomes the target of a melee attack. Rule #2 will apply, so the sector line will go into a controlled retreat, because Rule #2 has a higher priority than Rule #1. If the enemy retreats because they lost their will to fight, the condition in Rule #2 will stop being true, because the sector line is no longer the target of a melee attack (this condition is based on a local event, therefore it is not persistent). Since the event in Rule #1 is global, making it persistent, the sector line will go back to engaging the enemy in ranged combat.

Tactic

A tactic (or tactical plan) is nothing more than a formation created by the player, filled with the contingent categories he chooses (and where he chooses to place them), with a set of rules for each sector line in that formation. Tactics are created by players using the Tactical Planner.

Battle

A battle is a confrontation between two or more opposing armies, who are each lead by one general (i.e. one player). Each battle is simulated by the Battle Simulator. In order for that to happen, the simulator needs to know what armies are present on each side, and it also needs a tactical plan for each side with which to organize the available contingents in the battlefield and to give them their orders.

After each battle is simulated, its results are generated and stored so that the Battle Viewer can show them to the players. These results include every instant of the battle, so the player can see everything that happened during the battle and learn from it.

Fig. 24 shows a mapping of these concepts and how they are connected to each other. The player is represented in green. The Tactical Planner, Battle Simulator and Battle Viewer are represented in orange. The concepts related to the tactical planning are represented in blue, and the concepts related to the armies and battles are represented in yellow.
3.2 The Battle of Cannae in our Conceptual Model

We will now show how the Battle of Cannae can be translated into our Conceptual Model, by defining both Roman and Carthaginian tactical plans in our model. This will demonstrate that the Conceptual Model can reproduce this historical battle.

Roman tactical plan

The conventional deployment for armies of the time was to place infantry in the center and deploy the cavalry in two flanks. The Romans followed this convention fairly closely, but chose extra depth rather than breadth for their infantry line, hoping to use this concentration of forces to quickly break through the center of the Carthaginian army. (Healy, 1994)

According to Fig. 1, we can see that the Roman infantry (in red) had one thick line in the center of the formation, which we can reproduce in our Model by having two sector lines in the center and choosing Melee Infantry as the Contingent Category for all the slots in both sector lines. Notice that only setting one sector line with Melee Infantry in the center will have
the exact same effect as having two, in the battlefield, because sector lines have no depth limit.

The Roman cavalry, as can be seen in Fig. 1, is placed in both flanks, although the left flank has a higher density in contingents. We can reproduce this in our Model by placing Melee Cavalry in the front lines of both flanks and filling an extra slot in the second sector line of the left flank. The final formation is shown in Fig. 25, together with the number of soldiers that will be assigned to each slot during the battle.

![Fig. 25 – Roman formation in the Battle of Cannae](image)

Concerning the orders given to each sector line, we can see from Fig. 1 that the Roman tactics were to send the whole center in a Melee Attack as soon as the battle started, the typical tactic used in order to break the enemy lines as soon as possible. This means assigning the same Rule for each sector line in the center:

<table>
<thead>
<tr>
<th>Priority</th>
<th>Condition</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Battle : Start</td>
<td>Melee attack</td>
</tr>
</tbody>
</table>

The Roman cavalry, on the other hand, which was weaker and outnumbered by the Carthaginian cavalry, was ordered to defend until the Carthaginians retreated from the battlefield. Therefore, these are the orders that players would give to both Roman flanks in the tactical plan:

<table>
<thead>
<tr>
<th>Priority</th>
<th>Condition</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Battle : Start</td>
<td>Defend</td>
</tr>
<tr>
<td>2</td>
<td>Enemy : Center : Retreats</td>
<td>Pursuit</td>
</tr>
</tbody>
</table>

4 The two golden insignias on each slot represent its rank, which is Rank 2, or the Average Rank
Carthaginian tactical plan

Hannibal had deployed his forces based on the particular fighting qualities of each unit, taking into consideration both their strengths and weaknesses in devising his strategy. He placed his Iberians, Gauls and Celtiberians in the middle, alternating the ethnic composition across the front line. Hannibal’s infantry from Punic Africa was positioned on the wings at the very edge of his infantry line. These infantry were expertly battle-hardened, remained cohesive, and would attack the Roman flanks. (Healy, 1994)

According to Fig. 1, we can see that the Carthaginian infantry (in blue) was in the shape of a Double Echelon formation (an Echelon is a formation where the contingents are arranged diagonally). This can be reproduced in our model by using the two front sector lines in the center and the third on both flanks, placing Melee Infantry only on specific slots, in order to reproduce the Double Echelon shape as best as possible (see Fig. 26). The slots with Melee Infantry in both flanks should be placed with the highest rank in order to place the best soldiers there before the battle starts.

The Carthaginian cavalry, as can be seen in Fig. 1, should be placed on both flanks, with a higher density on the left. Therefore, two sector lines should be filled on the left flank, and only one on the right flank. The final formation is shown in Fig. 26, together with the number of soldiers that will be assigned to each slot during the battle.

![Carthaginian formation in the Battle of Cannae](image)

The Rules assigned for each sector line are more complicated than the ones assigned to the Roman army in order to reproduce the complex tactics that gave Hannibal his

---

5 The golden insignias on each slot represent its rank; three insignias for Rank 3, the Highest Rank
victory. For the sector lines in the center, the following table describes the Rules we should use:

<table>
<thead>
<tr>
<th>Priority</th>
<th>Condition</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Battle : Start</td>
<td>Defend</td>
</tr>
<tr>
<td>2</td>
<td>Group : This : Target of : Melee attack</td>
<td>Controlled retreat</td>
</tr>
<tr>
<td>3</td>
<td>Group : Opposing : Target of : Flank attack</td>
<td>Melee attack</td>
</tr>
</tbody>
</table>

For the flank’s infantry, the following table describes the Rules we should use:

<table>
<thead>
<tr>
<th>Priority</th>
<th>Condition</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Battle : Start</td>
<td>Defend</td>
</tr>
<tr>
<td>2</td>
<td>Group : Friendly : Center : Target of : Melee attack</td>
<td>Flank attack</td>
</tr>
</tbody>
</table>

For the flank’s cavalry, the following table describes the Rules we should use:

<table>
<thead>
<tr>
<th>Priority</th>
<th>Condition</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Battle : Start</td>
<td>Melee attack</td>
</tr>
<tr>
<td>2</td>
<td>Group : Opposing : Retreats</td>
<td>Rear attack</td>
</tr>
</tbody>
</table>

### 3.3 Platform Architecture

In this section we present the architecture for our platform to give the reader an idea of how it works and how its components are connected to each other (see Fig. 27). Both the Tactical Planner and the Battle Viewer, which are GUIs, were programmed using the Model-View-Controller (MVC) architecture described in Olsen (1998). This architecture separates the application semantics (the model) from how it is presented to the user (the view) and how the user interacts with it (the controller). In practice, both GUIs use the same model, but present it differently to players, because the first is intended to help players create tactical plans and the second is intended to show battles to players.

- The Game Config loads all the game information from various files in a configuration folder, when either the Tactical Planner, the Battle Simulator or the Battle Viewer need that information; this way, we can change the game elements (weapons, contingent categories, contingents, damage types) in one place and it affects the whole platform.
- The Tactical Planner allows players to create and save new tactics for their armies, as well as load existing ones. These are saved into a folder named “tactics”, which is stored in the game server.
• The Battle Simulator simulates all the battles on the server side (players do not interact with the simulator). When these battles are simulated depends on the rules defined by the game designers who use the platform.
  o It needs to load the tactical plans from both sides of that battle, as well as its armies. Each battle has an XML file associated to it on the “battles” folder, describing the armies on both sides and the tactical plans to be used. These XML files are currently created and edited by hand.
  o After simulating each battle, the battle itself (contingent movement, damage done, and so on) is stored in an XML file, which will be inside a folder called “results”. Each battle has its own XML file.
• The PHP Server is needed because this whole platform was developed in Adobe Flash, and storing any information from Flash requires connecting to a server, which then handles the necessary files.
• The Battle Viewer allows players to view battles that have already taken place, i.e. that have already been simulated by the Battle Simulator. The Battle Viewer loads a specific XML file from the “results” folder and presents that battle to the player.

![Diagram](image)

Fig. 27 – Our platform architecture
4 Tactical Planner

The Tactical Planner, which is our main Graphical User Interface (GUI), is the biggest concern in this work. It must allow players to create abstract tactical plans -- plans where there is no concrete army involved -- without disregarding the various categories of contingents available, their positions in the formation, the different orders given by the generals, when to execute those orders, and even the possibility to choose where to place the strongest or weakest contingents. All these features will allow for varied and rich tactical plans, to the point where players can even copy the ones from famous historical battles.

In order to reach a GUI that had a good usability, we created several prototypes and tested them with various subjects from different areas of expertise and academic degrees, different ages and levels of experience in interacting with personal computers and window-based applications. Most of the subjects had never played a videogame based on ancient or medieval tactics.

We started by making low-fidelity prototypes (LFP’s) in paper because, according to Preece (2007), they are easy, quick and cheap to make, and they don’t require any lines of code. Also, the LFP’s allowed us to see how the test subjects would interact with the Tactical Planner, how they expected it to react to their actions, and how easy the language used was to them.

As we tested each LFP and saw the difficulties our test subjects had, we felt the need to create new prototypes that improved usability (to make tasks easier for our subjects) and to evolve the Conceptual Model in order to use a language that was closer to that of the subjects. Also, we noticed that the early prototypes violated a few of Nielsen’s heuristics in Dix (2004), another reason why we needed to create new prototypes and avoid making the same mistakes. This process went according to the Iterative Design described by Preece (2007), which is shown in Fig. 28.

![Iterative Design of a Graphical User Interface](image-url)
The Conceptual Model also suffered some changes based on our demanding goals for the platform, because we wanted the player’s experience in creating tactical plans to be as close as the generals’ when they were preparing their armies for battle. Therefore, the reader will notice some discrepancies from prototype to prototype due to those changes. We only presented the most recent Conceptual Model in Chapter 3, and the tactical concepts in that model are used thoroughly in the last functional prototype.

This chapter presents our three low-fidelity prototypes (LFPs) in Section 4.1 and our two functional prototypes (FPs) in Section 4.2. For each of those prototypes, we describe its objective, how many test subjects there were and what task was asked of them, we show the evaluation results attained in each, what we learned from our mistakes and the solutions we found to improve the following prototypes. All the tests were observed from up close, because the test subjects usually required help to understand what was asked of them, since they were not familiar with the concepts. Observations were written down and in the end the test subjects were asked to fill a small survey (see Appendix B).

4.1 Low-Fidelity Prototypes

LFP #1

In the first low-fidelity prototype (see Fig. 29), the objective was to test how players would behave when asked to create groups of soldiers inside the grids that represent each sector in a formation — the center, the left flank, the right flank and the reserves. The two reserve sectors (shown at the bottom of Fig. 29) were removed after testing this prototype, as we decided to allow players to make some sector lines behave like reserves – the contingents placed in those sector lines would stay in their positions until something happened during the battle (an event) that made them follow their orders.

Each group in this prototype would be assigned a specific category (in the prototype we called it “group type”) and would consist of one or more blocks — the squares that together form each grid. Since each block is an area of 50x50 meters on the battlefield, there is a maximum number of soldiers that can be placed inside it (e.g. 250 infantry, 125 cavalry and 25 artillery units), thus making a group have a maximum number of soldiers as well. We wanted to see if players would take those numbers into account when creating their groups, because for this prototype players would know the number of soldiers in each of their contingents.

A Group is a concept that is not part of the Conceptual Model in Chapter 3; it was replaced by the Slot in the following prototypes. A group could actually correspond to one or more slots at the same time, depending on its width. On the other hand, nothing prevented players from creating a group in each block, using this prototype.
The width of each grid (and consequently of each sector) in terms of blocks is directly linked to the width of each sector in the Battle Simulator (see Chapter 5): each flank sector is 10 blocks wide and the center sector is 20 blocks wide. In this prototype, the depth for the center and the flanks was fixed (6 blocks deep) but, as we understood that a tactical plan should be available for any army, regardless of its size, we decided to have a variable depth for each sector, depending on the number of soldiers assigned to them.

Groups are created by clicking the “New Group” button (shown at the top of Fig. 29) and then dragging a rectangle over one or more empty blocks. Groups can also be resized or dragged to another part of the formation, as long as there is enough space for them (groups are not allowed to overlap in this prototype).

Clicking a group shows a panel with options regarding that group: the type of soldiers to be placed in it during a battle, their density (if they are in a tight or sparse formation) and the priority of that group (groups with a higher priority would be filled first). Group density and priority are concepts that are not part of the Conceptual Model anymore. The priority of a group was replaced by the ranks, which work similarly, but prevent entire groups in the formation from having no soldiers assigned when in a battle. The density of a group was removed as a choice for players.
LFP #1 Testing

Although each sector was made of a large amount of blocks, the goal was for players to create big rectangle-shaped groups for specific types of soldiers. Though we tested this prototype with three test subjects who are experienced with graphical interfaces, and they had no problem creating, resizing or moving groups, we found that having that many blocks in each sector was a bad approach. It complicated the user’s task, because the fact that each block had a maximum number of soldiers forced players to know how many soldiers they had in their contingents, which changed with each battle (because soldiers would die), and forced them to update the tactical plan after each battle to take into account the losses (and probably newly recruited contingents). We realized that we should avoid giving players all that work and instead create a tactical plan that would not take contingent size into account.

The options available for each group was something that two of the subjects did not understand, because they were not familiar with the concepts about ancient and medieval battles, and also because the panel was confusing and offered no help. The available group types, for instance, should not be split into two combo boxes because some of the combinations did not even make sense (e.g. Melee Artillery -- artillery is supposed to be fixed and to fire at distant enemies). Two of the test subjects did not know the difference between a tight or a loose formation, because they had not played any videogame of the genre. Also, the test subjects did not understand which priority was highest (1 or 10) -- they defined this scale as being ambiguous. To make matters worse, group options were not even visible outside the panel, which forced players to memorize what they had chosen for each group.

The group priority defined in this prototype, which is not contemplated in the Conceptual Model, was how we defined which groups were to be filled first by the available contingents in the army. This method could make low-priority groups completely empty during a battle, something players would not be expecting if they calculated group sizes wrong, and they would not like to see it happen in a battle. Hence, we decided to use ranks instead of priorities, which let players choose where to place their strongest (or weakest troops), while still assigning the same number of soldiers to groups of the same type, leaving none empty.

Initially we wanted each group to have a specific order but, considering the number of groups a formation could have (one per block), it would just become impractical to show all the information on the screen, and the visualization of the battle would not look realistic. So the testing for this prototype did not consider giving orders to groups.

LFP #2

The second prototype we designed was much simpler than the first concerning the number of blocks in each sector (see Fig. 30). The main idea was that, in the Tactical Planner, the number of blocks per sector had to be reduced to only a few, and the Battle Simulator would have to translate (i.e. convert) the choices made in those few blocks to all the available blocks inside the battlefield as best as possible. Details about this translation are explained in Chapter 5. We decided to change the name of the blocks in the Tactical Planner
to slots, since having the same name to describe two different things would be confusing for the reader and for the players.

As we learned from the first prototype, tactical plans – and thus, formations -- had to be army-independent, meaning players should be able to use the same tactical plan in different battles and with different armies, regardless of their size. This aspect had two benefits and one problem. The benefits were that (a) we could make a lot of different and usable tactical plans available for players, based on historical ancient and medieval battles, and (b) players would not need one tactical plan for each of their armies, nor to adapt it each time one of their armies engaged in a battle or merged with another one, because they could just use the same tactical plan for every army they controlled. The problem was that we had to give players the idea that the tactical plans they were creating were abstract, that they were not meant for a specific army and did not use any of those armies’ contingents.

In this prototype, the way in which players select the type of soldiers for each slot has changed. Instead of creating a group and then switching its category to the desired one, players now have a list of the available contingent categories on a panel (lower left panel in Fig. 30) and they only have to drag the desired category to the desired slot (or slots).

To replace the group priorities from the first prototype, we created ranks, which allow players to choose where to place their strongest (or weakest) soldiers. By using ranks, we ensure that slots of the same category will be assigned the same amount of soldiers, and the strongest soldiers will be assigned to the slots with the highest ranks, as well as the weakest soldiers to the slots with the lowest ranks.
Assigning a contingent category to a slot is as simple as dragging the desired category from the panel into the desired slot. Changing the rank of a specific slot is done by clicking on the rank icon on the top right corner of the slot. In this prototype, every slot starts at rank 1; clicking the icon changes the slot to rank 2, then to rank 3, then back to rank 1.

Assigning an order (in this prototype it’s called an “action”) to a sector is done by clicking on the designated button (below the name of the sector) and choosing the order that sector should follow from the menu that is shown. We have also added events to this prototype, because not all orders were followed right after the battle started. It was common in ancient and medieval battles to attack in waves (e.g. one line after the next), or to wait for something (an event) to happen during the battle. We wanted each order to be coupled with an event, so we added a list of events that we thought would be meaningful to players who were experienced in this genre. Events can be changed in the same way as orders are: players click on the corresponding button and then change the event to the one they want.

Another feature introduced in this prototype was the option to add, remove or switch lines (and columns) from a sector. Clicking on a slot outside the area where the category icon fit would show a menu with these options (see Fig. 30).

Our objective for this prototype was to make it easier (comparing to the previous prototype) for players to place the desired contingents on the formation, to change their ranks easily, and to give a specific order (and event) to each sector.

**LFP #2 Testing**

Eight subjects tested this prototype, half of which had no experience with videogames that involved ancient or medieval battles. Although these tests were done on paper, all the test subjects had some experience with computers, more specifically with a window-based operating system like Windows, which helped them to know what to expect from the prototype and how to interact with it. All of the test subjects performed the following task, after being shown the prototype with an empty formation:

1. Place swordsmen on the first row of the central sector
2. Place archers on the second row of the central sector
3. Place spearmen on the first row of both flank sectors
4. Place cavalry on the second row of both flank sectors
5. Remove the spearmen from both flanks
6. Move the cavalry on both flanks to the first row
7. Switch the center’s first row with the second row, leaving the archers in front and the swordsmen behind
8. Increase the rank of the archers in the middle of the first row
9. Change the order on the left flank to “Flank” when the “Shock phase” takes place during battle
10. Change the order on the right flank to “Surround” when the “Ranged phase” takes place during battle
11. Remove the third row from the central sector

The easiest task for all the test subjects was assigning a contingent category to a slot, which consisted of dragging the desired category to the desired slot. A few subjects even
suggested clicking the category once to select it, then clicking on the desired slots to assign that category to them, thus avoiding going back and forth with the mouse cursor.

Dragging a category from one slot to another was also an easy task for all the subjects, although moving entire lines (or columns) was something most subjects did not understand how to do, because the prototype did not highlight the desired line (or column), and instead they moved every group in that line (or column) to the other -- they were not doing it wrong, but it was the hard way. Our objective was for the testers to use the menu shown in Fig. 30, but they did not know it existed. When asked to remove the third row from the central sector (step 11), all of the test subjects already knew the menu existed (because they were told about it in step 7), so they knew how to perform that task now.

Switching the rank of a slot was not something we expected the least experienced test subjects to understand, because they were not familiar with the insignias we used as icons, and therefore they did not think about clicking on the icon. They thought dragging the same category on top of the desired slot (which was already assigned that category) would increase its rank.

Orders and events had the worst results during the tests, because the test subjects did not understand how to change the orders or what the events were for. The squares we used in the prototype did not resemble buttons, so the subjects did not know they could click on them. After telling the test subjects those squares were buttons and showing them the menu with the available orders and events, the images used to represent these orders and events were not familiar to most of the test subjects (even the experienced ones), so they had a hard time associating any of those images to the orders mentioned in their task -- the images should have come with a description. We learned that the buttons for the orders and events were not placed correctly. Since the first row was the top one on the prototype, the buttons should have been placed on top of that row to indicate the direction the soldiers were facing.

LFP #3

After testing the previous prototype, we concluded that having only one order per sector was too restrictive, since some ancient and medieval battles depicted in books were far more complex (e.g. the Battle of Cannae, which we describe in Section 1.2). With that in mind, we decided to divide each sector into lines and allow one order for each sector line (see Fig. 31).

We learned from the previous prototype that the orders should be put on top of the sector lines, since soldiers are supposed to be facing up (the top line is the first line in the formation, if we consider the battlefield), and the orders’ icons should help players recognize where the whole formation is facing.

Based on the tests made for the second prototype, we decided to place the available orders in the panel, together with the contingent categories (here called “groups”), because they are assigned to a sector line in much the same way as a category is assigned to a slot, so the way players do both tasks should be similar. Also, since orders are something every
player – from beginner to experienced -- must be able to use, they must be easily accessible. And since the test subjects had no problem dragging the contingent categories to the desired slots on the last prototype, assigning an order to a sector line should work the same way.

Since most of the test subjects had trouble when asked to change the rank of a slot, we decided to make ranks work the same way as categories: the available ranks should be visible on the panel and dragged to the desired slot. A slot will be assigned the default rank when a category is assigned to that slot, but players can change its rank at any time.

Adding, removing or switching lines inside a sector was something that was not very clear in the last prototype, and the test subjects did not guess how to do it, so we decided to remove that feature, forcing each sector to have two lines (or three, if necessary).

The number of slots per sector line, on the other hand, was made dynamic. A sector starts with only one empty slot and, when a player assigns a category to that slot, another slot is added to each side of the now assigned slot, as long as that side has no slots. When the number of slots in a sector line reaches its maximum, no more slots are added. The purpose of this dynamism is for players to avoid assigning all the slots in a sector line with the same contingent category, when they can assign only one and the outcome in the battlefield will be the same, because the whole line will be filled with contingents of that category.

For instance, if we consider Fig. 31, the first line in the center sector only has one slot filled with Ranged Infantry. Because the other two slots are empty, the Battle Simulator will assume that the whole line is to be filled with ranged infantry. But if players do not want the whole line to be filled, they must place the “red sign” on the empty slots, which leaves them empty in the battlefield.
Events are still placed below the corresponding sector line, as they were in the previous prototype, but now they are listed in a combo box, so players will understand how to use it. The default event for each sector line is set to “Battle start”, in case players forget to change it. Events are something we’re not expecting beginners to use, but they should allow some interesting choices for experienced players.

**LFP #3 Testing**

Eight subjects tested this prototype, half of which had no experience with videogames that involved ancient or medieval battles. Although these tests were done on paper, all the test subjects had some experience with computers, more specifically with a window-based operating system like Windows, which helped them to know what to expect from the prototype.

All of the test subjects were asked to recreate the tactical plan shown in Fig. 32 in the prototype. This new approach was based on the fact that, in the previous prototypes, dividing the whole task into subtasks was helping the test subjects by telling them what to look for, which could compromise the results.

The dynamic number of slots per sector line was confusing for all of the test subjects, so instead of easing the task it actually complicated it, because initially the subjects thought they did not have enough slots to place all the necessary contingent categories in the formation (the formation in Fig. 32 showed four slots per line, and subjects only had three in the prototype – one for each sector). The test subjects only noticed new slots showing up when they started placing categories in the available slots, and they were not pleased with that feature.

As we expected, having the orders on top of their corresponding sector lines helped the test subjects recognize the first line in the formation, because of the icons’ orientation —
this time, none of the test subjects dragged contingent categories into the bottom line slots, thinking it was the front line in the formation.

Just like in the previous prototype, placing groups inside the sectors (by drag and drop) was something no test subject had trouble with. Still, one of the subjects had trouble changing the orders given to sector lines, because the subject did not realize it worked the same way as for categories. The rank buttons were confusing for three of the subjects, because they were not of the same size as the category (in this case “group”) buttons, so those three test subjects assumed they worked differently.

Even with the changes made to the way players should handle the event for each sector line (by using the combo box), the test subjects still had difficulty in understanding what events were for and how to change them. Half of the test subjects did not know they had to use the combo box. That was partly our fault, because it already had a default value selected, so the test subjects assumed they did not need to change it. Although we are not expecting beginner players to use events, we are still finding it difficult to express its importance to the test subjects.

4.2 Functional Prototypes

FP #4

Since we were pleased with -- and confident about -- the results of the tests on the last prototype, we made prototype #4 functional because we wanted the test subjects to try it on a computer. This prototype (see Fig. 33), which was implemented in Adobe Flash, is similar to the previous one, so we will just list the differences.
The concept of having a dynamic number of slots per sector line, which was meant to reduce the time players spent designing their tactical plans, was discarded since it confused the test subjects. Each sector line now has the maximum number of slots it can hold and players cannot add or remove slots to any sector line.

To avoid players clicking every single slot in a sector line when they want to fill it with the same contingent category, we added an auto-complete feature to this prototype, which fills adjacent slots with the selected category, but only if those slots are empty. This feature is intended to reduce the time players spend designing their tactical plans.

One thing we realized, when testing the previous prototype, was that the test subjects were not using the combo boxes to change the events, because they thought the bottom panel was the only area they could interact with and, as such, the green area was only used to drop objects from the panel and to have an overview of the tactical plan.

Therefore, we decided to make everything work with drag and drop. We divided all the items into three groups: the Groups (where players can pick the Contingent Categories they want for their slots, as well as the Ranks), the Orders, and the Events (where players can define when a sector line executes its order). Information concerning the events is now shown with icons – instead of the combo-box text – next to the order given to each sector.

**FP #4 Testing**

Eight subjects tested this prototype, of which only one had experience with videogames that involved ancient or medieval battles, and four had minimum knowledge about tactics from movies or other media. All the test subjects had some experience with computers, more specifically with a window-based operating system like Windows.

Because we found no significant changes in how the test subjects performed their task in the previous prototype, when we gave them the image with the whole tactical plan and asked them to recreate it, we decided to go back to using a step-by-step task. The task for this prototype was as follows:

1. Place Ranged Infantry on the first line of the central sector
2. Place Melee Infantry on the second line of the central sector
3. Place any type of Cavalry on the second line of both flank sectors
4. Remove one slot of Ranged Infantry from the center
5. Move the Cavalry on the left flank to the first sector line
6. Increase the rank of the three middle Melee Infantry slots
7. Change the order of the Ranged Infantry sector line to Skirmish
8. Change the order of the Cavalry on the left flank to Flank Attack
9. Make the Cavalry on the left flank wait for the Ranged Infantry sector line to start shooting the enemy, so they can then start their Flank Attack

The eight subjects spent from five to seven minutes executing the task (with an average of six minutes) and made three mistakes on average, mostly related to changing the order for each sector line and dealing with events (we considered their giving up on the task...
as making a mistake). At the end of the task, the subjects gave the prototype a score of 3.9 / 5.0 in organization (where 1.0 is disorganized and 5.0 is organized), and 3.1 / 5.0 in simplicity (where 1.0 is difficult to use and 5.0 is easy to use). Tab. 2 shows how this prototype was scored by each test subject, along with its average score and standard deviation. This table was based on answers given to question 7 of Appendix B.

<table>
<thead>
<tr>
<th></th>
<th>Organization</th>
<th>Simplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>3.9 (max 5.0)</td>
<td>3.1 (max 5.0)</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>1.07</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Tab. 2 – Functional Prototype #4 score for organization and simplicity

As in the previous two prototypes, the test subjects had no problem placing contingent categories in any slots, moving them from slot to slot or removing them from their current slot, although four would have preferred dragging instead of clicking. The reason why we force users to click is because users associate the action of dragging with dragging only one object at a time, which forces them to do the same action repeatedly when they want to fill various slots with the same category. By clicking, we attach the desired category to the mouse cursor, and the user can then click as many slots as he wants, and the prototype will assign that category to the clicked slots. It works similarly to programs like Microsoft Paint and Adobe Photoshop.

This method, on the other hand, caused three of the subjects to become confused in how to discard the category associated to the mouse. They tried pressing the Escape key and right-clicking on the screen, which showed a Flash-specific menu with options. The correct action was to click on any other part of the screen that is not relevant to the tactical plan.

When asked to change the rank of the three middle Melee Infantry slots (step 6), two of the subjects tried clicking on the rank icon on top of the slot, until they noticed they had to use the rank buttons on the bottom panel. Three other subjects went for the rank icon on top of the slot, but noticed the panel buttons before clicking.

When asked to change the order of two sector lines, no subject had any problem noticing the Orders button. The problem was clicking on the right spot on the battlefield: three of the subjects tried clicking on any slot of the desired sector line when they had the order associated to the mouse cursor. The correct action was to click on the order icon associated to the desired sector.

All the test subjects had problems understanding the way events were represented in this prototype and how to use them. The idea we wanted to transmit in the prototype was that the order (shown as a white icon) would only be executed when the event (shown as one or two blue icons) happened. In Fig. 33, the Ranged Infantry in the first center line is ordered to Skirmish when the battle starts (this event is represented by the blue “1”), and the Cavalry to its left is ordered to make a Flank attack on the enemy as soon as the Ranged Infantry starts shooting arrows at the enemy (represented by the blue bow and arrow). Our test subjects did
not understand how to do this task, nor did they feel the prototype transmitted the right message concerning the event, when we showed them how it was done.

**FP #5**

The Functional Prototype #5 (see Fig. 34), which is the last prototype we designed, brought some noticeable changes. First of all, the lines in the formation are now drawn vertically (with its orientation from left to right), instead of horizontally (with its orientation from bottom to top), because screens have a much bigger width than height and we should take advantage of that space.

Although a formation with two lines might be suited for casual players, experienced players might have the need for more complex tactics, where they need extra lines in their formation, and our GUI should take that into consideration. Therefore, it is again possible to add and remove lines to the formation.

Most applications that use the method we are using for filling slots have their panel on the left side of the window, so we should be coherent and do the same. We are referring to the “click the desired button and click where you want to put it” method, instead of the dragging method. Programs like Microsoft Paint, Adobe Photoshop and Adobe Flash Professional are some examples of applications that use that method.

We noticed that most of the test subjects preferred clicking the rank icon over the slot, so we decided that is how they should work, but there is also a second reason: the rank is a property of the category on that slot, so it should be changed on the spot. It should not be picked from the panel and placed over the category.

![Functional Prototype #5](image)
The order icons on each sector line now have a square around it to indicate where the player should click to change the order. And there is now a “+” button near each order icon, when the sector line is not empty, which allows players to set additional orders for that sector line. Clicking on it shows a modal panel (see Fig. 35) where players can set up to five orders and four events for each sector line. The main idea for the design of this panel was based on the solution found in Dragon Age: Origins (see Section 2.6).

The first order in the panel (which is shown in the battlefield) is associated to the “Battle : Start” event, and that event cannot be changed because every sector line must at least execute one order when the battle starts. This was decided so that players know the first order their sector lines will follow when a battle starts. The default order is now “Defend”, instead of a “Frontal attack” (now “Melee attack”), because players may not notice the Attack order when assigning categories to their slots and end up losing contingents on an attack they did not expect to happen. This way, we avoid players making that mistake.

For the remaining orders, players can decide what event to associate them with, making the contingents on that sector line only follow those orders when those events happen during the battle.

**FP #5 Testing**

Eight subjects tested this prototype, of which only one had experience with videogames that involved ancient or medieval battles, and four had minimum knowledge about tactics from movies or other media. All the test subjects had some experience with computers, more specifically with a window-based operating system like Windows.
The task for this prototype was as follows:

1. Move the Melee Infantry on the first center line to the second center line
2. Place Ranged Infantry on the first center line
3. Place any type of Cavalry on the second line of both flanks
4. Remove one slot of Ranged Infantry from the center
5. Place Melee Infantry on the first line of the left flank
6. Increase the rank of the Melee Infantry slots on the left flank
7. Remove the third line from the formation, since it will not be used
8. Change the order of the Ranged Infantry sector line to Ranged Attack
9. Change the order of the Melee Infantry on the center to Melee Attack
10. Change the order of the Cavalry on the left flank to Flank Attack
11. Give a second order to the Cavalry on the right flank: make them wait for the Ranged Phase, so they can then start their Flank Attack

The eight subjects spent from three to seven minutes executing the task (with an average of five minutes) and made one mistake on average related to dealing with events (we considered their giving up on the task as making a mistake). At the end of the task, the subjects gave the prototype a score of 4.5 / 5.0 in organization (where 1.0 is disorganized and 5.0 is organized), and 4.0 / 5.0 in simplicity (where 1.0 is difficult to use and 5.0 is easy to use). Tab. 3 shows how this prototype was scored by each test subject, along with its average score and standard deviation, and shows the score of the previous functional prototype for comparison. This table was based on answers given to question 7 of Appendix B.

From these results, we conclude that the subjects’ appreciation of the new prototype was higher (compared to the previous prototype), that their making one less mistake was an improvement (also compared to the previous prototype), and that the average time (to finish the task) decreasing by one minute, coupled with the task having two more steps, shows this prototype was easier to use and to understand than the previous prototype.

<table>
<thead>
<tr>
<th></th>
<th>Organization</th>
<th>Simplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FP #5</td>
<td>FP #4</td>
</tr>
<tr>
<td>Average</td>
<td>4.5 (max 5.0)</td>
<td>3.9 (max 5.0)</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.53</td>
<td>1.07</td>
</tr>
</tbody>
</table>

Tab. 3 – Functional Prototype #5 score for organization and simplicity

The decrease in the number of mistakes was due to the square that was added around the order icon in each sector line. Without it, some of the test subjects did not know where to place the order icon carried by the mouse cursor. Now none of the subjects had any trouble changing the orders to any sector line.

We concluded that the one mistake that five of the subjects made, which was mostly due to giving up on setting the second order to the cavalry (step 11), was because the language on this prototype still needs to be improved: most players are not familiar with it
when they first start, and we should take that into consideration. Of the three subjects that did manage to set the event and the second order, one of them is experienced in this videogame genre and the other two hesitated at first, but managed to accomplish the task.

The remaining steps were done without any problem: assigning categories to slots, moving them around, removing them, and changing the rank of various slots. It is important to notice that three of the test subjects still tried to drag the categories to the desired slots, instead of clicking on them and then clicking on the slots, just like they did with the previous prototype.

Another change we decided to add to this prototype, after testing, was to allow players to press the Escape key to discard the category (or order) icon attached to the mouse cursor, because three of the test subjects used that key in this prototype and in the previous. So now there will be two ways to discard the icon: clicking on the background or pressing the Escape key.
5 Battle Simulator and Battle Viewer

Although the Tactical Planner is the most important part of this work, it would be incomplete if there was no way to show players how their tactical plans worked out in a battle. Therefore, the platform also includes a Battle Simulator and a Battle Viewer, which use some of the ideas referred in Section 2.7, where we compared how the various videogames showed each battle to players. The way the Battle Simulator works is also based on Appendix A.

The aim of this whole work is to allow players to create tactical plans for their armies before the battles take place, giving them no control over their contingents during the battle. Because we needed a way for the contingents to follow their orders autonomously, we chose to implement the Battle Simulator as a MultiAgent System (Wooldridge, 2002), where each agent controls one contingent in the battlefield. Another reason was because we wanted to create agents with simple rules and see if their emergent behaviors would help create complex and realistic battles.

The Battle Viewer will show battles in 2D, seen from the top. The contingents will be represented as rectangles with a shape on top describing their categories, and each side will have its own color (red or blue), as is usual in books that depict ancient and medieval battles. Contingents will also show their current action so players know what each one is doing at all times. Contingent movement will have no accelerations or decelerations, but rotations and collisions will be present. On the main window, players will see a specific area of the battlefield from a point of view, which players can move around at any time, and there will also be a minimap showing a complete view of the battlefield and the position of each contingent.

Section 5.1 introduces the battlefield as the environment in which the agents (i.e. the contingents) live in, and explains the properties of this environment. Section 5.2 describes formations in a simulation context, shows where each army will be placed in the battlefield, and presents the algorithm that places one or more armies in their designated area, using the formation defined on their tactical plan. Section 5.3 describes contingents in a simulation context as agents, their internal state and the sensors and actuators they use to interact with the environment they live in. Section 5.4 describes the Battle Simulator, the process of simulating a battle and the storing of relevant data for later viewing. Section 5.5 describes the Battle Viewer and the way it shows battles to players.

5.1 Battlefield: The Environment

The battlefield is the environment in which the agents (i.e. the contingents) live and interact. Currently, this environment is as simple as a static grassy plain (of approximately 5.4 x 5.4 kilometers) where the contingents are placed and follow their orders.
The reason why we limit the dimensions of the battlefield is due to retreating contingents (contingents that are running away from the battlefield). In real battles, the winning side would sometimes pursue its enemy to either make them prisoners or to slaughter them; in some cases, the losing side would manage to escape unharmed. We want to give the winning side a chance to catch enemy contingents that are retreating, so we made the battlefield big enough for the contingents to take a while to escape. If a retreating contingent does manage to reach the border of the battlefield, it survives the battle but is considered as “retreated”, which means it cannot participate in the current battle anymore.

According to Wooldridge (2002), the environment for each battle will be:

- **Accessible**, meaning any agent can obtain complete, accurate, up-to-date information about the environment’s state
- **Non-deterministic**, meaning any action performed by an agent may not have a single guaranteed effect (e.g. a contingent might try to move to a certain point in the battlefield, but be unable to do so because it collides with another contingent)
- **Dynamic**, because while one or more agents might not do any action in the environment, the remaining agents will and those actions will change the environment
- **Discrete**, meaning there will be a fixed, finite number of actions and perceptions in it.

### 5.2 Placing contingents in the battlefield

Before a battle begins, the Battle Simulator must load the tactical plans and armies pertaining to that battle, and all contingents (on both sides) must be placed in the battlefield and must be given their orders. Placing contingents in the battlefield is done according to the algorithm described throughout this section, which makes use of the formation defined in the corresponding tactical plan. We call the contingents in an army “Army Contingents” (AC), the contingents in a slot “Slot Contingents” (SC) and the contingents in the battlefield “Battle Contingents” (BC).

#### Step 1: Assigning contingents to slots

The first step of the algorithm consists in assigning each of the contingents in the army (ACs) to the most suited slots in the formation chosen for the battle. The algorithm

- organizes all the non-empty slots in the formation by category (Melee Infantry, Ranged Infantry, and so on) and orders them by rank, from strongest to weakest
- organizes the available ACs by category and orders them by cohesion -- which translates the overall value of an AC into a number -- from highest to lowest
Having both slots and ACs divided by category and ordered from strongest to weakest, it becomes easy to assign the strongest ACs to the slots with the strongest ranks and the weakest ACs to the slots with the weakest ranks.

The total number of soldiers in each category must be split evenly among all the slots of that category. We start by adding the number of soldiers of each AC from that category, and then dividing that total by the number of slots of the same category. That will tell us how many soldiers should be placed in each slot. Then we assign as many soldiers as possible from the first AC (the strongest) to the first slot (with the highest rank) without surpassing the limit for each slot. Then we place as many as we can on the second slot without surpassing the limit, and so on, until the first AC has no soldiers left. We repeat this for the next AC, and the next, until all ACs are assigned a slot.

Fig. 36 helps explain this step. Suppose we have 4 Melee Infantry ACs making a total of 3000 soldiers, and 10 Melee Infantry slots -- for this example, it does not matter where these slots are in the formation. Remember that slots and ACs are already ordered from strongest to weakest. Each slot will be assigned 300 soldiers. AC 1 has 850 soldiers, so slots 1 and 2 will each be assigned 300 soldiers from AC 1. Slot 3 will be assigned the remaining 250 from AC 1 and 50 from AC 2 -- there is no problem in assigning different ACs to the same slot. We repeat the same process for the remaining slots and ACs.

Just to be clear, AC 1 will be split into three identical Slot Contingents (SCs), the only difference being the number of soldiers on each: two SCs with 300 soldiers and one SC with 250 soldiers.

The algorithm must also consider that players can forget to use some categories in their tactical plans. If we consider that a player creates a tactical plan without assigning any slots for Ranged Cavalry, and then uses that tactical plan on a battle where his army has Ranged Cavalry ACs, those ACs should also be placed in the battlefield. Since there are no slots for Ranged Cavalry, the ACs should be assigned to slots of another category, one that is best suited for Ranged Cavalry.

In some videogames, there is the concern of assigning roles in a specific formation to a group of characters, each character with a specific role. Obviously, a character with role A should be assigned to a spot in the formation that is for characters of role A. The problem is how to deal with the assignments when there are no more characters with the required role for the remaining spots.

The solution found was to use “soft roles” (Millington, 2006): instead of a character having a role (or a list of roles) it can fulfill, it has a set of values representing how difficult it is...
for that character to fulfill every role. The ideal role for a character should have zero cost. For our work, the characters are our ACs and the spots in the formation are our slots. Table 4 shows how hard it is for each AC category to fulfill the category of each slot. The algorithm will always assign each AC to the first available category with the lowest cost.

Because infantry and cavalry had different roles and speeds in ancient and medieval battles, we decided that the priority should be to assign an infantry contingent to an infantry slot and a cavalry contingent to a cavalry slot. Only then should the algorithm assign army contingents to slots based on their weapon type (ranged weapon or melee weapon). Artillery, which had a completely different role than infantry and cavalry, has the highest cost, to avoid placing infantry and cavalry in artillery slots, and artillery in infantry or cavalry slots. That should only happen if there is no other choice.

<table>
<thead>
<tr>
<th>Contingent</th>
<th>Slot</th>
<th>Melee Infantry</th>
<th>Ranged Infantry</th>
<th>Melee Cavalry</th>
<th>Ranged Cavalry</th>
<th>Artillery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melee Infantry</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Ranged Infantry</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Melee Cavalry</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Ranged Cavalry</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Artillery</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Tab. 4 – Weighted table for placing any army contingent in any slot

Let us consider a new example to contemplate the costs in Tab. 4, which can still be visualized in Fig. 36. Assume that we still have the 10 Melee Infantry slots, and that we now have a Melee Infantry AC with 850 soldiers, a Ranged Infantry AC with 450 soldiers and low cohesion, another Ranged Infantry AC with 1300 soldiers and high cohesion, and a Melee Cavalry AC with 400 soldiers.

The first thing the algorithm will do is organize the ACs by category, and then by cohesion: the ACs will be organized into three categories, and the Ranged Infantry ACs will be ordered by cohesion. Then the algorithm must assign the 10 slots to these ACs, based on the costs on Tab. 4. The Melee Infantry AC has cost 0, so it will be assigned first (AC 1 in Fig. 36). Then the Ranged Infantry ACs will be assigned next, because their cost is 1 (AC 2 and AC 3 in Fig. 36). Finally, the Melee Cavalry AC (with cost 2) will be assigned last (AC 4 in Fig. 36). Even if the Melee Cavalry AC has a higher cohesion than the Melee Infantry AC, it will still be assigned last, because its cost is the highest.

**Step 2: Calculating the number of blocks needed for each slot**

Blocks are the smallest element in the battlefield. Currently, the battlefield is nothing more than a grid with 100 x 100 blocks, and each block is 54 x 54 meters in size, hence the battlefield being 5.4 x 5.4 kilometers in size.
Fig. 37 gives the reader an idea of the battlefield size and of the areas each side has for their contingents to be placed: each little square is a block and the red and blue rectangles are the slots in the formations. In this figure, each slot is 5 blocks wide and 4 blocks deep, to give the reader an idea of the various elements and their dimensions in the battlefield. However, neither depth nor width of each slot is fixed, it all depends on the number (and size) of SCs assigned to them. The only restriction is that a slot cannot be more than 5 blocks wide.

Inside the battlefield, Battle Contingents (BCs) are always 50 meters wide and have a variable depth, depending on the number of soldiers they have, but their depth never exceeds 50 meters. This means each BC in the battlefield has a maximum number of soldiers:

- 2000 infantry: 40 lines with 50 soldiers each, spaced every meter
- 500 cavalry: 20 lines with 25 horsemen each, spaced every 2 meters
- 100 artillery, 10 lines with 10 artillery each, spaced every 5 meters

Each block is intended to carry one full BC (50 x 50 meters), and the blocks’ extra 4 meters in width and depth are strictly for every BC to move around freely, avoiding unnecessary collisions with nearby BCs.
After assigning the ACs to the slots, consequently turning them into SCs, the second step of the algorithm is to calculate how many blocks each slot needs, so the algorithm knows the slots’ dimensions in the battlefield (number of blocks in width and in depth). For this, the algorithm goes through all the SCs assigned to each slot and divides the number of soldiers in each SC by the maximum number of soldiers a BC of that category can have. The sum of all the divisions will be the number of blocks needed for that slot and, consequently, the number of BCs that slot will have in the battlefield. Table 5 shows an example.

<table>
<thead>
<tr>
<th>SC 1 (Melee Inf.)</th>
<th>SC 2 (Ranged Inf.)</th>
<th>SC 3 (Artillery)</th>
<th>SC 4 (Melee Cav.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of soldiers</td>
<td>5000</td>
<td>1000</td>
<td>13</td>
</tr>
<tr>
<td>Maximum per block</td>
<td>2000</td>
<td>2000</td>
<td>100</td>
</tr>
<tr>
<td>Number of blocks required</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total number of blocks</td>
<td>3 + 1 + 1 + 4 = 9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab. 5 – Number of blocks required for a slot with four assigned contingents

**Step 3: Aligning the slots horizontally and vertically**

Slots in the Tactical Planner are aligned horizontally and vertically (see Fig. 34), and we want the same to happen in the battlefield when placing the BCs. This does not mean that all the slots must have the same width or depth, because many times that will not be true since some slots will be assigned more soldiers than others. If they all had the same depth and width, we would not need to align them.

Based on the battles depicted in books, we can see that contingents standing side by side in the battlefield are always aligned. We can apply this idea to the lines in our formations: all the BCs in each slot will be placed in the battlefield starting with the front blocks and moving to the rear blocks, and all the slots in each sector line will have their front blocks aligned.

Fig. 38a shows an example of a formation with two lines, and each slot in that formation has a number determining how many blocks that slot needs in the battlefield to place the SC (or SCs) assigned to it. Remember that the only restriction is that slots cannot be more than 5 blocks wide. For this example, it does not matter which SCs were assigned to each slot, only the number of blocks each slot needs.

Fig. 38b shows the same formation from Fig. 38a, but now it only shows the center lines with all their slots aligned horizontally. Notice that the first and third slots in the first sector line are two blocks deep, because they cannot be more than 5 blocks wide.

In order to align the slots in the second line, we must take into account the deepest slot in the first sector line (i.e. the last line of blocks it used), so that the front blocks in the second sector line are placed after that last line of blocks used. Considering Fig. 38b, if the first sector line needs two blocks for its depth (due to the first and third slots), then the slots in the second sector line can only start in the third line of blocks.
Besides aligning slots horizontally, we also need to align them vertically. The way the algorithm handles this problem is by looking for the first slot in each “column” (for lack of a better word) with BCs assigned, checking its width (in blocks) and assigning that width to the remaining slots in that “column”. Fig. 38c shows an example, based on the sector lines in Fig. 38b.

Fig. 38d shows an example considering that the second slot in the first sector line has no SCs. Notice that the second “column” takes the width of its second slot, because it is not empty. In case all the slots in a “column” are empty, the algorithm will assume the player left them empty on purpose, and assign all the slots in that “column” the maximum width (in this case, 5). In the end, the formation will have a hole there, in the battlefield, leaving the nearby “columns” vulnerable in their flanks (i.e. their sides). If the readers can imagine no colored blocks in the second “column” in Fig. 38d, that will give them an idea of a formation with an empty “column”.
5.3 Battle Contingent: The Agent

“An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.” (Wooldridge, 2002)

In our work, we use reactive agents with state (see Fig. 39) to control battle contingents, more specifically one agent per contingent. We have decided to use reactive agents with state for the following reasons:

- Since every agent controls a battle contingent, agents must keep the state of their contingents in order to execute the best order for them (e.g. if the contingent loses its will to fight, the agent must retreat from the battlefield).
- Since there are various orders an agent can follow, from moving to ranged or melee combat, agents must know which order they are supposed to be following, and what they are actually doing at any given moment (e.g. the agent is supposed to engage the enemy in melee combat, but it is still moving towards the nearest enemy contingent).

This section refers the attributes of a battle contingent, how we model an agent’s behavior, how agents move in the battlefield, the damage types available for them and how they choose their targets.

![Diagram](Fig. 39 – A reactive agent with state)

**Contingent Types**

Our platform comes with a list of contingent types ready to use in any battle and more can be added by the game designers. Each contingent has attributes that define its role in the battlefield and some of those attributes, like the ones listed below, are defined when a new contingent type is created, and their values depend on the desired type of contingent.
- **Name**, the name of the contingent type (e.g. Swordsman, Longbowman, Knight)
- **Speed**, how fast the contingent moves in the battlefield (cavalry contingents are the fastest, heavy infantry contingents are the slowest, and artillery contingents do not move)
- **Ranged weapon**, the type of ranged weapon the contingent carries
- **Ranged damage**, how much damage the contingent does with its ranged weapon
- **Range**, how far the contingent can throw their arrows (or other projectiles) at their enemies
- **Melee weapon**, the type of melee weapon the contingent carries
- **Melee damage**, how much damage the contingent does with its melee weapon
- **Protection**, how much armor the contingent has (more protection means taking less damage from a weapon)
- **Morale**, the base morale of the contingent, which affects its cohesion (cohesion will be explained later)
- **Category**, used to assign the contingents to the best possible slot in the formation, when placing them in the battlefield

**Army Contingents**

Every contingent in an army has a specific type, which already defines some of its attributes, as was shown in the previous sub-section -- “Contingent Types”. Additional attributes regarding the contingent’s current situation in an army are listed below:

- **Number of soldiers**, how many soldiers the contingent currently has
- **Health points**, how much health each soldier in the contingent has
- **Combat experience**, the combat experience of the soldiers in the contingent
- **Status**, how tired the soldiers on the contingent are (the higher the number, the more rested they are, and the better they are in battle)
- **Cohesion**, defines how long a contingent holds on in battle before losing its will to fight

Cohesion is a complicated concept, so we will explain it here. According to (Henderson, 1985), military cohesion is defined as “the bonding together of members of an organization/unit in such a way as to sustain their will and commitment to each other, their unit, and the mission”. The more prepared soldiers are for battle, the higher their cohesion will be. A contingent’s cohesion is calculated based on the following formula:

\[
Cohesion = (Base\text{Morale} + Protection + Damage + \sqrt{XP}) \ast (0.2 + 0.4 \ast Status)
\]
The contingent’s morale defines its will to fight, so the higher the better. The heavier the armor of a contingent, the safer its soldiers feel, so it improves their cohesion in combat. The better the weapons of a contingent, the more damage they deal, so it improves the contingent’s cohesion. The more experienced a contingent is in combat, the more it has learned to be cohesive. Tiredness (i.e. status) also affects the cohesion of a contingent: tired soldiers are weakened soldiers and, as such, more prone to losing their will to fight.

Behavior

In our Battle Simulator, agents’ behavior is defined by a state machine. Agents in a game usually act in one of a limited set of ways and they will carry on doing the same thing until some event or influence makes them change. State machines are the best technique to represent this behavior, as they were designed for it. (Millington, 2006)

In our work, the state machine for every agent is fairly simple (see Fig. 40), because there are not that many actions they can do: agents can move around the battlefield towards a target area (or enemy), defend their position (a special case of movement where the agent’s speed is set to zero), shoot arrows (ranged combat), shock against an enemy contingent, be in melee combat, or retreat from the battlefield.

Fig. 40 – Agent state machine for the Battle Simulator
When a battle contingent is placed inside a block in the battlefield and assigned an agent to take control over its actions during the battle, that agent is given the set of Rules that the player assigned to the sector line where that battle contingent was placed. It is these Rules that will define the agent's behavior (i.e. its actions) during the battle. Every agent starts with their state machine in the Moving State.

The orders that players can give their contingents (in their tactical plans) only affect their movement (i.e. their Moving State). Players do not order their contingents when to engage the enemy in ranged or melee combat, they simply tell them to execute a frontal attack (melee or ranged), a flank attack or a rear attack, for instance, which consist of telling them from which direction to attack the enemy. The “Melee Attack” order will make the contingent move towards its nearest target, the “Flank Attack” order will make the contingent move around the battlefield to attack the enemy from its flank, and so on. The moment when they actually attack the enemy depends on when they get in range for it, which the player has no control of.

Apart from the “Battle Start” event, all the other events will be triggered by the agents in the battlefield, whether they are in their ranged state, their melee state or retreating. When those events occur, agents may be assigned a new order (according to their tactical plan) and their movement may change due to that new order. This means their Moving State will tell them to move to a new point in the battlefield, because their order changed.

If an agent is in any state except the Moving State and a new event occurs, the order associated to that event will have no effect on the agent’s current action -- meaning the agent will not change its current state -- until the agent returns to its Moving State because, as was said, orders only affect the agent’s Moving State.

The exception is the “Retreat” order, which is of the highest priority, because the player wants to rescue the remaining soldiers in the battle contingent. When a contingent retreats, either by the player’s order or because it lost its cohesion, the player loses control over it, meaning the contingent will no longer follow the orders assigned to it in the tactical plan.

Choosing a target

Every agent updates its target regularly, depending on its current order, its surroundings and its current situation.

When the agent is executing a flank attack, its target is a point in the battlefield close to the enemy’s left or right flank (whichever is closest), until the agent reaches that spot or finds an enemy contingent along its way that has its flank unprotected.

When the agent is retreating, it tries to move away from nearby enemy agents, to avoid getting more of its soldiers killed. The closer the enemy agent is, the more the retreating agent tries to avoid it.

When the target is not supposed to be a point in the battlefield, but rather an enemy agent, there are a number of factors that determine which enemy agent is the best target.
These factors have weights that are put together and multiplied, and the enemy agent with the highest value is assigned as the new target. Table 6 shows the various factors and weights used to determine an agent’s new target.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance to the enemy agent (D)</td>
<td>1 / D</td>
</tr>
<tr>
<td>If the enemy agent was our agent’s last target</td>
<td>1.5</td>
</tr>
<tr>
<td>If the enemy agent is routing and its speed is higher</td>
<td>0.1</td>
</tr>
<tr>
<td>If there is no clear view to enemy agent</td>
<td>0.5</td>
</tr>
<tr>
<td>If the enemy agent is routing and our agent is made up of infantry</td>
<td>0.5</td>
</tr>
<tr>
<td>If the enemy agent is in ranged combat and attacking our agent</td>
<td>3.0</td>
</tr>
<tr>
<td>If the enemy agent is in melee combat and attacking our agent</td>
<td>10.0</td>
</tr>
<tr>
<td>If our agent is in melee combat and attacking the enemy agent</td>
<td>5.0</td>
</tr>
<tr>
<td>If our agent is in ranged combat and attacking the enemy agent</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Tab. 6 – Factors and weights for determining an agent’s new target

As an example, let us consider a situation with four agents (see Tab. 7): one agent from the “blue” army (B) and three agents from the “red” army (R1, R2 and R3). Agent B is in melee combat with agent R1. Agent R2 is moving towards agent B, and they can both see each other. Agent R3 is shooting arrows over agent R1, hitting agent B -- this means that agent B cannot see agent R3. Based on Tab. 7, we can calculate the new target for agent B by multiplying the various weights specific to each “red” agent. The next chosen target will still be R1, which was also the last chosen target because they were already in melee combat.

<table>
<thead>
<tr>
<th>Factor</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance to the enemy agent (D)</td>
<td>1 m</td>
<td>100 m</td>
<td>50 m</td>
</tr>
<tr>
<td>Distance weight (1 / D)</td>
<td>1.0</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>If the enemy agent was our agent’s last target</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If the enemy agent is routing and its speed is higher</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If there is no clear view to enemy agent</td>
<td></td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>Enemy agent is routing and our agent is made up of infantry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enemy agent is in ranged combat and attacking our agent</td>
<td></td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td>Enemy agent is in melee combat and attacking our agent</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our agent is in melee combat and attacking the enemy</td>
<td></td>
<td></td>
<td>5.0</td>
</tr>
<tr>
<td>Our agent is in ranged combat and attacking the enemy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>75</td>
<td>0.01</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Tab. 7 – Calculating a new target for the “blue” agent
Movement

Most of what any contingent will be doing during a battle is moving around to reach its goal. Combat itself, be it melee or ranged, does not take as long as moving around the battlefield towards an enemy contingent or towards the edge of the battlefield (when retreating). The only orders that were integrated from the Conceptual Model (which affect a contingent’s movement) were the “Melee Attack” order, the “Ranged Attack” order, the “Skirmish” order, the “Flank Attack” order, the “Retreat” order and the “Defend” order, which already allow for varied tactics and battles.

The remaining orders in the Conceptual Model were not implemented in the Simulator for the following reasons: the “Pursuit” order, because it is already implicit in the “Melee Attack”, the “Ranged Attack” and the “Flank Attack” orders, since the contingents following those orders will keep pursuing retreating enemies; the “Flank Attack” order will be turned into a “Rear Attack” order in some situations, so the latter becomes redundant; the “Fake Retreat” and the “Controlled Retreat” orders were not so commonly used in ancient and medieval battles, and they made more sense when used with local events, which we also did not integrate in the Simulator.

Collisions are an important aspect in the simulation as well, because soldiers in the same battle contingent were usually close to each other, for cohesion purposes, making it difficult for one battle contingent to cross another. Every time an agent makes a move in the battlefield, the Battle Simulator checks if that move is allowed (which means it checks if that agent is not colliding with another, be it friend or foe). There are two types of collisions: between friends and between foes.

Between friendly contingents, the only times that collisions are not checked is when one (or both) of them is either skirmishing or retreating. This is because soldiers would keep some distance from each other when skirmishing (allowing them to move through other contingents towards their rear, for safety), and because when they were retreating in panic, they spread and lost their cohesion.

When two enemy contingents collide, a shock starts between the two (the shock phase) and eventually they engage in melee combat, after the shock phase is over. The only time a collision is not checked between two enemy contingents is when they are both retreating because, having lost their will to fight, they will not engage each other in combat.

Damage

Because there were different types of weapons in the ancient and medieval ages, we needed to define different types of damage to the simulation. Two of them are very common in videogames nowadays: melee damage (caused by swords, axes and maces) and ranged damage (caused by bows, longbows and crossbows). We also added another damage type that is not so common, but which can be seen in Total War, especially when a cavalry contingent makes a charge: shock damage.
Shock damage is dealt when two opposing contingents charge towards each other and finally collide, making it the bloodiest phase in hand-to-hand combat. The front lines in both contingents are crushed against each other, a lot of those soldiers die in the process, and after a while the remaining lines enter melee combat, where they try to find some space to swing their weapons at their enemies.

The damage inflicted by any weapon is calculated using the following formula:

\[
WeaponDamage = BaseDamage \times \sqrt{XP} \times (0.5 + 0.5 \times \text{Status}) \times \left(0.5 + 0.5 \times \frac{\text{ActualCohesion}}{\text{InitialCohesion}}\right)
\]

Inside a contingent, soldiers are divided into lines and the front lines inflict more damage than the rear ones, depending on the weapon’s length. Soldiers in the rear lines have a hard time reaching the enemy with their weapons, because they are further apart. Therefore, the damage each contingent does to another must take this aspect into consideration.

Let us consider a contingent with 5 lines and 10 soldiers in each line. If each soldier in this contingent has a weapon damage of 20, and if we consider their damage is decreased by 10% for each line (100% damage in the first line, 90% damage in the second line, and so on), then the final damage for that contingent will be:

\[
\begin{align*}
FinalDamage &= \sum_{n=1}^{5} (1.1 - 0.1 \times n) \times 10 \times \text{WeaponDamage} \\
&= 200 + 180 + 160 + 140 + 120 = 800
\end{align*}
\]

This final damage is multiplied by a factor, depending on the type of damage being inflicted: shock, melee or ranged. These factors also depend on the weapon types used by both contingents in the conflict, and they can vary from 0.5 to 2.0.

When a contingent is inflicted some physical damage, its defense is applied to that damage in order to reduce it: having better armor and more health points will reduce the damage taken. The final damage inflicted to a contingent, after the defense was applied to reduce it, is calculated using the following formula:

\[
\text{Casualties} = \frac{\text{FinalDamage}}{\text{HP} + \text{Protection} \times \text{WAP}},
\]

where WAP is how good the armor is against the weapon inflicting the damage

Shock damage, melee damage and ranged damage are all physical damages, but we must also account for the soldiers’ psyche. Soldiers would start to lose their will to fight when they saw their friends dying or retreating from the battlefield. Therefore, we must consider the
morale of each contingent, because it was a very important factor in battles. That is why we introduced the concept of cohesion.

When a contingent inflicts physical damage to another, it also inflicts cohesion damage, because soldiers start dying and the others start losing hope from seeing their friends die. The only form of cohesion damage that is not physical is when a contingent watches another one retreating. A contingent that is retreating causes cohesion damage to nearby friendly contingents, because seeing soldiers retreating makes the others want to retreat as well.

Cohesion damage is calculated by multiplying the weapon damage by a factor, depending on the type of damage inflicted: shock, melee or ranged. These factors depend on the weapon types used by both contingents in the conflict, and they can vary from 0.5 to 2.0.

### 5.4 The Battle Simulator

Each battle has a battlefield and a simulator associated to it, so the platform allows for various battles to be simulated simultaneously on the server side (see Section 3.3). Each battlefield (the environment) is associated to a simulator and that simulator must load the correct armies and tactical plans belonging to that battle in order to place the contingents in the battlefield (see Section 5.2) and allow the battle to take place. After that, the simulator is basically a cycle that covers all the agents (i.e. battle contingents) in the battlefield and gives them a chance to analyze their environment (i.e. the battlefield and the other agents) and decide what to do, according to the orders assigned to them in their tactical plans. The battle state, which consists of the contingents’ positions, actions and status, is stored after each step of the cycle is complete for later viewing on the Battle Viewer.

Agent actions are divided into two categories: moving and dealing damage. Any moving action affects the environment instantaneously, as long as there are no collisions, in which case the action has no effect. Any damage that agents inflict on each other is only processed at the end of the cycle, after all the agents have had a chance to act, to maintain balance. For instance, if agent A attacked agent B and agent B suffered the damage right away (let us say some soldiers would die in that attack), then the damage agent B inflicted back to agent A would be reduced because it had less soldiers to inflict the damage. Therefore, to keep the balance, the simulator was made so that each agent attacks its current target(s) and its target(s) store the damage inflicted to them, and only after the cycle is finished will the damage be applied to the targets’ status.

A battle ends when all the contingents from one side are either destroyed or outside the battlefield area -- when a contingent leaves the battlefield area, it is automatically removed from the battle. When the battle ends, the cycle (which covers all the agents) stops and the battle results are stored, so the Battle Viewer can then show the battle to the players. Battle results consist of a list of all the contingents present in that battle, along with their casualties. The winning side of the battle is also referred in the battle results.
5.5 The Battle Viewer

The Battle Viewer is a GUI that allows players to watch the battles in which their armies participated. When the Battle Viewer loads a battle, the player is shown a list of all the contingents on both sides of that battle, together with information like their category, number of soldiers and combat experience (see Fig. 41).

Players can then start watching the battle. When the battle is over, the player is shown statistics about it, like the casualties each contingent suffered and which side won the battle. The rest of this section introduces the various elements present in this GUI. Fig. 42 shows a picture of the Battle Viewer with all the elements listed throughout this section.

Battle playback

Ancient and medieval battles were long, so we decided to speed the visualization by a factor of 20: every second in real time corresponds to 20 seconds of a battle. That way, a one hour battle (3600 seconds) can be seen in three minutes (180 seconds) without the risk of the player losing any valuable information. There is a visible timer on the GUI that shows players the instant of the battle they are looking at, in “battle time” (not real time).

Players have a few playback options at their disposal: they can hit “play” or “pause”, or use the slider to move to a specific moment of the battle. This allows players to watch the battle as if it was a video. These options are situated at the bottom of the window (see Fig. 42), like in most video players (e.g. Windows Media Player, YouTube, QuickTime Player).
Since the main view of the battlefield only shows a small part of the battle, players can move the main view around the battlefield using the mouse -- by clicking and then dragging in the desired direction.

![The Battle Viewer showing an ongoing battle](image)

**Battle Contingents**

Agents (i.e. battle contingents) are represented by rectangles in the battlefield. Since there are two sides in each battle and the background is green, we decided to use blue for one side and red for the other. These are also the colors that books depicting ancient and medieval battles use to paint each side in the battlefield. To distinguish the five categories -- melee infantry, ranged infantry, melee cavalry, ranged cavalry and artillery -- we created a shape for each. Fig. 43 shows these shapes in the specified order, from left to right.

![Category shapes for contingents (facing right)](image)

The size of a rectangle depends on the number of soldiers in the contingent that rectangle corresponds to: the width of the rectangle will always be the same, but its depth
varies according to the number of lines those soldiers fill. This said, if a contingent suffers casualties, the depth of the rectangle will decrease and its category shape will scale as well.

There is also a small square on the right half of each contingent that indicates that contingent’s current cohesion: green if its cohesion is above 50%, yellow if it is between 25% and 50%, and red if it is below 25%. This helps players recognize when contingents are about to retreat from the battlefield.

Contingent information

By placing the mouse cursor over a contingent, players can see important information about that contingent, like the number of soldiers it has at that moment, its cohesion, category, combat experience, current action and speed. This information will be available in a panel that is shown close to the desired contingent. Fig. 44 shows an example of this panel.

The current action of a contingent is represented on top of the corresponding rectangle: they can be defending their position, moving, shooting arrows, shocking against an enemy contingent, in melee combat or retreating. Fig. 45 shows contingents performing all these actions, as they are shown in the Battle Viewer.
Notice that contingents that are retreating look like a chessboard. This symbolizes that those contingents can walk through other contingents (friend or foe), because its soldiers are not in a tight formation anymore. Retreating soldiers lose their tight formation, because they are in panic. Contingents that are skirmishing (shooting arrows and moving backwards to avoid physical contact with the enemy) are also shown like a chessboard, because they can walk through friendly contingents when moving backwards. This was a common tactic used by generals: the front line was filled with skirmishers, who would soften the enemy’s front line, and then a second line with melee soldiers would walk towards the enemy’s front line to engage them in melee combat while the skirmishers moved to the rear and were kept safe from harm.

**Minimap**

Because the whole battle cannot be seen in the main view of the battlefield, there is a minimap to give the player an idea of the whole battlefield and where the contingents from both sides are. Each contingent is represented by a dot of its respective color. The minimap also shows the main view of the battlefield, as a white rectangle, which allows players to know what part of the battlefield they are currently watching. The minimap is shown on the top right corner of the screen, as shown in Fig. 42.

![Minimap](image)

*Fig. 42 – The event panel showing an event being triggered during a battle*

**Global Events**

In order to help players understand the various phases in a battle, there is a panel on the bottom left corner of the screen (see Fig. 42) showing the global events that we describe in the Conceptual Model (see Section 3.1). When each of these events is triggered during the battle, its corresponding icon in the events panel will glow, warning the player that something important has happened in the battle. Fig. 46 shows the moment when the event “Battle : Melee Phase” starts. Notice that the event “Battle : Start” is different from the others (it has a
white background, instead of a transparent one) to show that the event was triggered in the past.

**Zooming**

Players may feel the need to zoom in on a specific part of the battle, or zoom out and have an almost complete view of the battlefield and all that is happening. Zooming in or out can be done by use of the mouse wheel, and the white rectangle on the minimap will resize according to the actual zoom, so players always know what portion of the battlefield they are watching in the main view. Also, there is a zoom bar at the bottom right corner of the screen (see Fig. 42) that tells players how close the view is from the ground.

**Battle Balance**

The Battle Viewer comes with a panel that shows players how many active contingents of each category are fighting on each side, how many are retreating, and what the battle balance is – if the red side is winning or losing the battle. This panel can be found below the minimap, and Fig. 47 shows an example of the panel.

![Fig. 47 – The battle balance panel](image)

The number on the left of each column is the number of active contingents, while the number in parenthesis is the number of retreating contingents. The bar on the top represents the battle balance, meaning it shows which side is winning the battle; it does not necessarily mean that side will win the battle.
6 Playtesting the platform

This chapter describes the playtest we prepared for the test subjects, to see how well they interacted with the Battle Viewer and the Tactical Planner, if they understood what was happening during a battle and if they learned from it. We had fifteen subjects test our platform, with ages ranging from 19 to 46, and two of the subjects were female. Ten of the subjects were not very experienced in battle tactics or games of the genre, and the remaining five were. Subjects 6 through 10 are the experienced ones (see Fig. 48).

We divided the subjects into three groups of five: Group1 consisted of inexperienced players who were given no explanation about the platform unless they asked; Group2 consisted of the five experienced players, who were also given no explanation about the platform; and Group3 consisted of another five inexperienced players who were given an explanation about the platform during the test, much like a videogame tutorial would.

We divided the subjects into these three groups to see how different the results would be between experienced and inexperienced players, and between inexperienced players who were given no explanation about the platform versus inexperienced players who were given that explanation.

The decision to explain the concepts in the platform, or to leave the concepts unexplained, was based on Schell (2008) and Fullerton (2008). Both methods are common in playtesting and achieve different results. Based on these books, we also decided to watch the subjects from up close, answering any questions they had, and we asked the subjects to fill a survey at the end of the playtest (see Appendix H).

The playtest

The playtest assigned to the subjects consisted of two steps. The first step involved the subjects interacting with the Battle Viewer and watching an already simulated battle where the blue army lost, due to having a slightly smaller army and having some inexperienced soldiers. The second step involved the subjects interacting with the Tactical Planner and creating a new tactical plan for the blue army, one that they thought would help the army win.
Fig. 49 – Formations for both armies in the playtest’s battle

Fig. 49 shows both armies’ formations, which consisted of placing the whole army in a single line, with all the swordsmen in the center, followed by the archers and skirmishers on both sides, and the knights on both flanks. The orders were for all the contingents to move forward at the start of the battle. Tab. 8 shows the contingents on both armies, as well as the number of soldiers on each and their combat experience.

<table>
<thead>
<tr>
<th>Contingent</th>
<th>Number of soldiers</th>
<th>Combat experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swordsmen</td>
<td>6 000</td>
<td>10 000</td>
</tr>
<tr>
<td>Archers</td>
<td>3 000</td>
<td>3 000</td>
</tr>
<tr>
<td>Skirmishers</td>
<td>3 000</td>
<td>0</td>
</tr>
<tr>
<td>Knights</td>
<td>1 000</td>
<td>1 000</td>
</tr>
</tbody>
</table>

Tab. 8 – Armies present in the playtest’s battle

From Tab. 8, we can see that the blue army had four thousand swordsmen less than the red army, and compensated with only three thousand skirmishers (a hybrid between an archer and a swordsman) who were not very experienced in combat. This required the subjects some thinking for the new tactics.

After creating the new tactical plan, the subjects watched the same battle but with the blue army following the newly created tactics; the armies remained the same, as did the red
army’s tactics. Finally the subjects evaluated the new battle results and determined if the second battle had a better outcome (for the blue army) than the first.

The playtest’s results

Fig. 50 shows some results about how well the three groups interpreted the first battle in the Battle Viewer.

![Graphs showing results](image)

Fig. 50 – Results showing how well the three groups interpreted the playtest’s first battle (based on questions 6, 7, 8 and 9 of Appendix H)

From Fig. 50a, we can see that Group1 was the group with the most difficulty in distinguishing the various contingents in the battlefield. Group2 had no difficulty recognizing the contingents, and in Group3 only one subject had difficulties. The reason why Group1 had such difficulties was because the subjects were not familiar with the way contingents are shown in the Battle Viewer, which is similar to books that depict ancient and medieval battles, and because the subjects had no introduction to the battle, like the subjects in Group3.
From Fig. 50b, we can see that Group2 and Group3 noticed the events happening during the battle (shown in the Battle Viewer’s Event Panel), while Group1 only noticed some of the events. Again, because the subjects from Group1 had no introduction to the battle, and to the existence of battle events, the panel remained unnoticed.

From Fig. 50c, we can see that thirteen subjects understood the various actions performed by the contingents during the battle: moving, shooting arrows, melee attacks, defending and retreating.

From Fig. 50d, we can see that thirteen subjects found the contingents’ movements to be acceptable: how they moved towards their targets, how they turned, how they retreated, and how they moved in formation.

All fifteen test subjects distinguished retreating contingents from active contingents (because of their chessboard effect). Of all fifteen subjects, only one (from Group1) needed to watch the first battle a second time in order to understand everything that had happened; every other subject watched the battle only once.

Tab. 9 shows us how our test subjects felt about their interaction with the Battle Viewer. The various elements and ways to interact with the Battle Viewer are listed in the first column, and values range from 1 to 5, where 1 is “hard to use / understand” and 5 is “easy to use / understand”.

<table>
<thead>
<tr>
<th>Battle Viewer - Interaction</th>
<th>Group1 AVG</th>
<th>Group1 STD</th>
<th>Group2 AVG</th>
<th>Group2 STD</th>
<th>Group3 AVG</th>
<th>Group3 STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving around the battlefield</td>
<td>4,25</td>
<td>0,50</td>
<td>4,40</td>
<td>0,55</td>
<td>4,80</td>
<td>0,45</td>
</tr>
<tr>
<td>Zooming in and out</td>
<td>4,80</td>
<td>0,45</td>
<td>4,80</td>
<td>0,45</td>
<td>4,80</td>
<td>0,45</td>
</tr>
<tr>
<td>Watching the status of a contingent</td>
<td>3,25</td>
<td>0,96</td>
<td>3,60</td>
<td>1,14</td>
<td>5,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Tracking the status of a contingent</td>
<td>3,33</td>
<td>0,58</td>
<td>3,20</td>
<td>0,84</td>
<td>4,67</td>
<td>0,58</td>
</tr>
<tr>
<td>Contingents Panel</td>
<td>3,00</td>
<td>1,22</td>
<td>4,25</td>
<td>0,50</td>
<td>4,80</td>
<td>0,45</td>
</tr>
<tr>
<td>Bottom Panel</td>
<td>4,80</td>
<td>0,45</td>
<td>5,00</td>
<td>0,00</td>
<td>5,00</td>
<td>0,00</td>
</tr>
</tbody>
</table>

Tab. 9 – Results showing subjects’ interaction with the Battle Viewer

From Tab. 9, we can conclude that the subjects from Group3 had the best experience (marked in green), since they were told what they could do with the Battle Viewer. The values from Group1 and Group2 are high in “Moving around the battlefield”, “Zooming in and out” and the “Bottom Panel” (marked in blue) because those interactions were necessary in order to watch the battle. The Contingents Panel had the lowest value in Group1 (marked in orange) because the subjects were not introduced to the Battle Viewer, and therefore did not understand all the information shown about the contingents. Watching and tracking the status of a contingent had average values (marked in red) because the subjects found the contingents hard to follow (and click) with the mouse cursor, due to the battle speed.
In the Tactical Planner, all the test subjects needed to be told that the tactics were independent of any army. It was something that was not obvious without some prior information. Seven of the subjects suggested adding a contingent category for the skirmishers, because having them in the same category as the archers was confusing. All the test subjects understood what each category was for, except for one subject from Group 1, who only understood the use for some of the categories.

Fig. 51 shows the results from the test subjects’ interaction with the Tactical Planner.

From Fig. 51a, we can see that eleven of the subjects understood what each order was for. The subjects from Group 3 understood all the orders because they were told what each order was for (just like a tutorial in a videogame would).

From Fig. 51b we can see that nine of the test subjects did not use the order buttons from the left panel on the Tactical Planner. Subjects from Group 1 and Group 2 had a curious and unexpected behavior: because they were not told how to use the Tactical Planner, they rarely assigned categories to empty slots, they only used the slots present in the tactical plan from the first battle and moved them around the formation. Also, when they wanted to change the orders assigned to each sector line, they used the panel where they can assign the five rules for each sector line, instead of using the orders on the left panel (see Appendix G).
From their behavior and based on Fig. 51b, we figured that the left panel was not very noticeable unless we told the subjects it was there.

Fig. 51c shows how many subjects used the ranks in the Tactical Planner. The army involved in the playtest’s battle was not very diverse, so there was not much need for using ranks, unless the subjects wanted to separate the archers (using a higher rank) from the skirmishers (using a lower rank).

Fig. 51d shows how many subjects used the events available in the Tactical Planner. Group1 was not told how to use the events, and when the subjects asked about the events they had some trouble understanding how the events would work in a battle, so only two of the subjects used them in the new tactical plan. Group2 had an easier time understanding the events, because the subjects were used to those concepts and figured out how to use them. All the subjects from Group3 were told how to use the events, so they used them in a way or another, mostly to keep some sector lines waiting for the right time to ambush the enemy.

When using the events, the subjects from Group3 were asked to think out loud, and their lines of thought were curious: they never thought about using global events, because those events were too unpredictable. What they wanted was for their contingents to react to events that happened in certain sectors of the formation. This meant using local events, which were not implemented.

It was curious to notice that all the playtest subjects used the Tactical Planner much more naturally than the prototyping subjects. The main reason for it was that the playtest subjects had a concrete task in their hands, which was to win the battle, while the prototyping subjects had a dummy task just to evaluate the prototype’s usability.

![Fig. 52 – Results showing subjects’ appreciation of the second battle](image-url)
Fig. 52 shows the appreciation of the test subjects after watching the second battle, the one where their tactics were put to the test. Fig. 52a shows that most of the subjects felt the agents were following the orders they were given most of the time, or all of the time. The two subjects from Group3 who did not like the agents’ behaviors used very complex tactics, giving every sector line three or four rules to follow, thus creating unexpected results due to the unpredictability of global events. The one subject from Group2 who did not like the agents’ behaviors expected the agents to execute the “Flank Attack” and the “Defend” orders differently.

Still, Fig. 52b shows that thirteen of the subjects found the second battle to have a better outcome than the first, even if in some cases their tactical plan was not executed as expected.

<table>
<thead>
<tr>
<th>Application</th>
<th>Group1</th>
<th>Group2</th>
<th>Group3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AVG</td>
<td>STD</td>
<td>AVG</td>
</tr>
<tr>
<td>Battle Viewer (first battle)</td>
<td>7,60</td>
<td>0,89</td>
<td>7,80</td>
</tr>
<tr>
<td>Tactical Planner</td>
<td>6,40</td>
<td>1,52</td>
<td>8,00</td>
</tr>
<tr>
<td>Battle Viewer (second battle)</td>
<td>6,00</td>
<td>0,71</td>
<td>6,40</td>
</tr>
<tr>
<td>Total</td>
<td>20,00</td>
<td>2,24</td>
<td>22,20</td>
</tr>
</tbody>
</table>

Tab. 10 – Time spent on each application during the playtest

Tab. 10 shows how much time was spent on each application during the playtest. Group3 took three more minutes (on average) watching the first battle because we explained the Battle Viewer to the subjects, just like a videogame tutorial would. In the Tactical Planner, Group3 took twice as much time as Group2 (on average) because they were told how to use it and they took some time preparing their tactics carefully. Group2 took almost two more minutes (on average) compared to Group1, because the subjects in Group2 were experienced in these types of games and therefore planned their tactics more cautiously. The time spent watching the second battle was very similar to all the groups, since it did not require learning anything new. Some battles were longer than others, depending on the tactics, but the average time spent was very similar.
7 Conclusions

The objective of this work was to create a platform for ancient and medieval battles, where players could plan their tactics before the battles took place (using a GUI), and then watch those battles after they were simulated (using another GUI). Our intention was for the tactics to be independent of any army’s composition, so they could be used by more than one army and in more than one battle. During the simulation of each battle, players would have no control over their armies, so every battle had to be resolved using a MultiAgent System.

The biggest challenge in this work was the prototyping of the Tactical Planner, because we needed the test subjects to understand what their role was (as a general), what they were allowed to do with their tactics and what effect their decisions would have in the battlefield. We were pleased with the improvements the test subjects showed, as we tested the various prototypes, especially with the last one. The events were by far the hardest concept to grasp, followed by the ranks, but we believe that we reached a good solution according to the results.

As for “rich and varied tactical plans”, having control over the number of lines in the formation, where to lay the contingents in the formation, where to place the strongest or weakest contingents, what orders to give to each sector line and what events they should react to, gives players a wide range of possibilities. Although we did not implement all the orders and events referred in the Conceptual Model, the ones that were implemented already allow for complex tactics and rich battles.

As for reproducing historical battles, part of our research focused on reading about historical battles to understand the types of decisions generals took, so that the platform would allow players to reproduce them. Our contingent categories cover all the types of contingents that existed in ancient and medieval times, although the playtest results suggest players would prefer having more concrete categories, especially when it comes to skirmishers, who are a hybrid between melee and ranged infantry. The orders in the Conceptual Model contemplate every choice a general could make, as far as our research allowed, but not having local events implemented in the platform (at the moment) reduces the number of historical battles we can reproduce with the Battle Simulator.

According to the playtest’s results, we concluded that Group3 had the best results because they were told how the applications worked and how to use them. Even Group2, composed of experienced players of the genre, did not have as good results because they did not know how to use all the features in the platform. Although the results were satisfactory in general, the ones from Group3 showed us that we need to have tutorials and graphical examples explaining how to use every feature in the Battle Viewer, and especially in the Tactical Planner, so players can enjoy the full scope of the platform and become true generals.
7.1 Future work

**Battlefield features**

In the future, it may be interesting to add some features to the battlefields, so players have a new level of complexity to worry about or use to their advantage when planning their tactics. Currently, we only have one battlefield implemented that is composed of grassy plains. In the future, however, we might want to add low hills, high hills, mountains, rivers, lakes, forests, desert areas, snowy areas, swamp areas, among other features, which will all have their influence in a battle.

One of the interesting aspects about using slots in every formation is that, depending on the features of the battlefield in which a battle takes place (like ponds or rocks), some of those slots may not be available to place the contingents, and the Battle Simulator can easily compensate for that by using the remaining slots in the formation.

Terrain features will require the implementation of path-finding algorithms in the Battle Simulator: rivers and lakes will have to be avoided, and mountains or high hills may not be the best path to follow, because they slow down soldiers.

**Weather conditions**

Weather conditions are something that may also be interesting to add to the platform, in the future, although this may bring some problems in terms of gameplay: the adding of rain would drastically decrease the damage done by arrows, as well as its range, and players need to be aware of that before the battle takes place; the adding of snow would make soldiers move slower and also decrease the damage done to the enemy; the time of day in which the battle took place, which influences the position of the sun, could also have an impact on a soldier’s performance – soldiers would see almost nothing with the sun in their eyes.

**Battle description in a single image**

Being able to describe a battle in a single image, like we saw in some of the books referenced in our work, is something that could allow players to have an overall view of the entire battle without having to watch it, saving them a lot of time. This idea will not make the Battle Simulator obsolete, because it will still be needed to simulate the battle in order to create the image. The image will need arrows showing contingent movement during the battle, the contingent positions at the end of the battle, how many soldiers each one had, as well as some additional information regarding the battle’s key moments. All of that information can only be given by running the Simulator, interpreting the battle status at each moment, filtering what is important and, finally, using all the information gathered and making an image that is clear and easy for players to understand.
References


**Other References**


Appendix A – Simulation Specifications

Side Characteristics
1. Rout Army Breakpoint (RAB) – Players define this

Contingent Characteristics
1. Base – Contingent type
   a. Basic Function (Foot, Cavalry, Art)
   b. Armor (Type, Weight, Protection)
   c. Shield (Weight, Protection)
   d. Health Points – HP
   e. Ranged Weapon (Type, Range, Damage - BD)
   f. Melee Weapon (Type, Damage - BD)
   g. Race modifiers for terrain

2. Secondary
   a. Quantity ( max foot: 250, cav:125, Art: 25 ) –QT
   b. Experience XP
   c. Status ST
   d. Unit battle readiness BR

3. Derived
   a. Terrain Suitability TS
   b. Shock efficiency CSE, HPSE
   c. Formation efficiency FE
   d. Cohesion CH
   e. Speed (m/min) SP
   f. Facing

Battle Terrain
1. Divided in squares (or use pure vectorial and line representation), roughly 50m x 50m
   a. Flanks: 10
   b. Center: 20
   c. Reserve Boxes: 8 x 5
   d. Army distance: 15-25
   e. Setup deep: 4-6
   f. Two possibilities:
      i. Center and Flanks have each abstract and uniform terrain characteristics, depending on the battle territory, with the center having the clearest terrain – More adaptable and versatile.
      ii. Individual terrain features in the map – esthetically better

Weapon Damage
Casualties - Units maintain density and frontage. Reduce depth.

Battle Resolution

1. Set up army
   a. Three sectors are independent, and the player may divide its contingent types per sector.
   b. Create default setups
   c. For movement purposes, group contingents per type within each sector

2. Order assignment
   a. One per sector / Unit Group OR
      i. Individual orders per contingent
      ii. Individual orders per contingent type
   b. Units in Reserve box have reserve priority orders (Off only, Off priority, mix, Def Priority, Def Only)

3. Each Turn = 5 min (all units at the same time)
   a. Move contingents (group by type) / Fire Missile
   b. Resolve Shocks
   c. Melee
   d. Retreat and/or Rout

4. ZOC's
   a. ZOC extends 20 m in front and 50 m to each side
   b. Enemy stops when entering ZOC, to fight frontal combat
   c. If there is already face-to-face combat with the enemy unit generating the ZOC, friendly unit may ignore the ZOC if it is not pinned with another overlaying ZOC
   d. Unit in enemy ZOC can only fight, change facing to face enemy, or Flee

5. Move/Fire missile
   a. Keep formation with units with same order while no contact with enemy (adjusting speed as necessary)
   b. Missile units with enemy in range, halt movement and fire
   c. Units with skirmish order
      i. Can move backwards at half the speed
      ii. fire and move backwards if enemy is too close (half the range)
   d. Unit Group that have been shot in this or in the previous turn move at a slower speed
e. Units move towards enemy
f. Units avoid obstacles and difficult terrain (regarding the various TS) – light troops will go over rough terrain, but not heavy troops.
g. If unit is in side/rear contact with enemy, and has no enemy in front, it turns to face it.
h. Units that shoot part of the turn (e.g. start their move outside of firing range) inflict proportional damage.
i. Ranged Damage
   i. Final Damage
   ii. Defender Casualties
   iii. Cohesion damage

6. Shock
   a. It happens when units contact each other for the first time
   b. Shock Damage
      i. Final Damage
      ii. Defender Casualties
      iii. Cohesion damage
   c. Unit inflicts Shock and Cohesion damage only to the unit fighting in their ZOC's.
   d. Unit fighting several different enemy units in its ZOC divides damage by percentage of ZOC occupied.

7. Melee
   a. Units fighting enemies in subsequent turns after shock, fight in melee.
   b. Melee Damage
      i. Final Damage
      ii. Defender Casualties
      iii. Cohesion damage
   c. Units inflict damage to all enemy units in melee. Damage is divided by percentage of contact.

8. Rout
   a. When units attain cohesion 0, they break, and start moving to the rear at 150% speed.
   b. A unit that routs provokes cohesion damage to friendly units up to a certain distance.
   c. Routed units inflict 0.25 * physical damage to other units when they collide
   d. When percentage of total initial army cohesion points exceeds RAB, all contingents rout. Reserve defending units will form a skirmish screen to try to protect retreating units and delay enemy.
Appendix B – Tactical Planner’s Survey

1. Do you have any experience with Windows (or another window-based operating system)?
   - Yes
   - No

2. If so, have you ever played a videogame where you had to make a tactical plan for an ancient or medieval army?
   - Yes
   - No

3. Are you familiar with the kinds of tactics generals used in our history (from the History Channel, medieval movies…)?
   - Yes
   - No

4. How old are you?
   i. 13 - 20
   ii. 21 - 30
   iii. 31 - 40
   iv. 41 - 50
   v. 51 - 60
   vi. 61 - 70

5. What is your academic degree?
   i. 9th grade
   ii. 12th grade
   iii. Bachelor’s Degree
   iv. Master’s Degree
   v. PhD

6. What is your profession?
7. What was your first impression about the Tactical Planner?

1 2 3 4 5

Disorganized Organized
Complicated Simple

8. For the following aspects of this Graphical User Interface, where there any difficulties found in/when…

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>…recognizing the first line in the formation (i.e. the front line)?</td>
<td></td>
</tr>
<tr>
<td>If so, why?</td>
<td></td>
</tr>
<tr>
<td>…filling (or emptying) the various slots with the available contingent categories?</td>
<td></td>
</tr>
<tr>
<td>If so, why?</td>
<td></td>
</tr>
<tr>
<td>…changing a contingent category from one slot to another?</td>
<td></td>
</tr>
<tr>
<td>If so, why?</td>
<td></td>
</tr>
<tr>
<td>…changing the rank of a certain slot?</td>
<td></td>
</tr>
<tr>
<td>If so, why?</td>
<td></td>
</tr>
<tr>
<td>…adding or removing lines to the formation?</td>
<td></td>
</tr>
<tr>
<td>If so, why?</td>
<td></td>
</tr>
<tr>
<td>…changing the main order given to a sector line?</td>
<td></td>
</tr>
<tr>
<td>If so, why?</td>
<td></td>
</tr>
<tr>
<td>…changing the events a sector line reacts to, and what orders it follows on those circumstances?</td>
<td></td>
</tr>
<tr>
<td>If so, why?</td>
<td></td>
</tr>
</tbody>
</table>

Thank you for your cooperation!
Appendix C – Prototype #1
Appendix E – Prototype #3
Appendix G – Prototype #5
Appendix H – Playtesting Survey

Player Information
1. Age: _____
2. Gender:
   a. M
   b. F
3. Experience in games / videogames containing ancient or medieval battles
   a. None (0)
   b. Reduced (1)
   c. Average (2)
   d. High (3)
4. Knowledge about ancient or medieval tactics
   a. None (0)
   b. Reduced (1)
   c. Average (2)
   d. High (3)

Battle Viewer – First Battle
5. How many times did the player watch the battle until he understood what happened?
   ________________________________________________________

6. Did the player feel any difficulty in distinguishing the various contingents (swordsmen, archers and cavalry) on the battlefield?
   ________________________________________________________

7. Did the player understand when the various events were triggered on the first battle?
   ________________________________________________________

8. Did the player understand the various actions performed by the contingents?
   ________________________________________________________

9. On average, how did the player feel about the movement of each contingent?
   a. Realistic
   b. Acceptable
   c. Unrealistic

10. Did the player notice (and understand) the difference between contingents that were fighting and contingents that were retreating?
    ________________________________________________________

11. At what distance did the player watch the battle?
    a. Far away
    b. At the default distance
    c. Close to the contingents
12. How was the interaction with the Battle Viewer?

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Difficulties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Moving around the battlefield</td>
<td></td>
</tr>
<tr>
<td>Zooming in and out</td>
<td></td>
</tr>
<tr>
<td>Panel with the global events (on the bottom left)</td>
<td></td>
</tr>
<tr>
<td>Watching the status of a contingent (mouse over)</td>
<td></td>
</tr>
<tr>
<td>Tracking the status of a contingent (click)</td>
<td></td>
</tr>
<tr>
<td>Contingents Panel (beginning and ending of the battle)</td>
<td></td>
</tr>
<tr>
<td>Bottom panel (playing, pausing, dragging the slider)</td>
<td></td>
</tr>
<tr>
<td>Understanding the action icons</td>
<td></td>
</tr>
</tbody>
</table>

Tactical Planner – Improving the Tactical Plan

13. Did the player…

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>... understand that the tactical plan is independent of any army?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>... understand what each order will do?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... understand what each contingent category is for?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... assign any orders to be executed after a battle event?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... change the rank in any slot?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... use the orders from the main panel? (the one on the left)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Battle Simulator – Watching the new battle

14. After creating the tactical plan, did the contingents execute their orders as the player expected them to?
   a. All the time
   b. Most of the time
   c. Sometimes
   d. Rarely
   e. Never

15. Did the player find the new tactical plan to have a better outcome in this battle?