The General Lord – Designing And Implementing A Persistent Multiplayer Mode in a AAA game

Marcos Pedrosa Pêgo

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Pedro Alexandre Simões dos Santos
Prof. João Miguel de Sousa de Assis Dias

Examination Committee

Chairperson: Prof. António Manuel Ferreira Rito da Silva
Supervisor: Prof. Pedro Alexandre Simões dos Santos
Member of the Committee: Prof. Rui Filipe Fernandes Prada

November 2021
This work was created using \LaTeX\ typesetting language in the Overleaf environment (www.overleaf.com).
Acknowledgments

I would like to thank my dissertation professors Pedro Santos and João Dias for the opportunity to work and participate on this challenging project and for their guidance and insights during the development of this project, as well as their help in shaping up the project by giving me constant feedback and helping me improve.

A very big thanks to my parents not only for their constant support and encouragement throughout these years, but also for being a cornerstone in my life and helping me ensure I concluded my final year in academics. Without them I wouldn’t have come as far as I did and probably wouldn’t even be able to enter or finish my degree.

I would also like to thank my family and friends for their endless support and encouragement. And for every participant that helped test my project and for their amazing feedback.

And lastly, a special thanks to all my colleagues Joaquim Quadrado, David Ricardo, Filipe Silveira and Marco Cabral, for sharing this hard journey with me and for their constant mutual help that helped all our thesis become a better project overall.

To each and every one of you, I thank you wholeheartedly for your support and friendship.
Abstract

This thesis documents the process and development of a modification to the triple-A Mount & Blade II: Bannerlord which aims to demonstrate the concept of adding persistent elements to the multiplayer experience. The end goal is to have a multiplayer experience with single player elements.

In an initial stage, research and documentation was analysed on existing games to analyse commonly used mechanics and systems that implement the relation between single-player persistent elements to a multiplayer experience.

Then a fully functional prototype was implemented on top of the existing game to create a brand new game mode, where the players were able to create their own personal army and test them out against other players.

The mod had shortcomings, but was successful in creating an enjoyable multiplayer experience with persistent elements in the game Mount & Blade II: Bannerlord. Users rated the mod rather high in terms of usability and enjoyment; it did however lack a good retaining factor for newer players which means there is still future work that could be done for the development and improvement of the mod.

Keywords

Mount & Blade II: Bannerlord · Gameplay Progression · Meta-Game Progression · Multiplayer
Resumo

Esta tese documenta o processo e desenvolvimento de uma modificação ao jogo triple-A Mount & Blade II: Bannerlord, que tem como objetivo demonstrar o conceito de melhorar a experiência multi-jogador com elementos persistentes. O objetivo final é ter uma experiência multi-jogador com elementos de único jogador.

Na fase inicial, foi analisada documentação em jogos existentes para analisar mecânicas frequentemente usadas e sistemas que implementem a relação entre mecânicas de único jogador a uma experiência multi-jogador.

Posteriormente foi implementado um protótipo completo e funcional assente no jogo já existente para criar um modo de jogo novo onde os jogadores pudessem criar o seu próprio exército pessoal e testá-lo contra outros jogadores.

O mod teve alguns contratempos, mas foi bem sucedido em criar uma experiência agradável de jogo multi-jogador com elementos persistentes no jogo triple-A Mount & Blade II: Bannerlord. Os utilizadores avaliaram o mod com valores altos em termos de usabilidade e agradabilidade; no entanto falta-lhe um bom fator de retenção para novos jogadores, o que quer dizer que ainda existe margem para trabalho futuro que poderá permitir a melhoria do mod.

Palavras Chave

Mount & Blade II: Bannerlord · Progressão de jogo · Progressão persistente · Multi-jogador
Contents

1 Introduction .................................................. 1
  1.1 Motivation ................................................. 3
  1.2 The Problem ............................................... 4
  1.3 Goals ...................................................... 5
  1.4 Results .................................................... 5
  1.5 Outline ................................................... 5

2 Related Work .................................................. 7
  2.1 Game Progression Theory ................................. 9
    2.1.1 Game Mechanics ........................................ 9
    2.1.2 Experience Duration .................................... 10
    2.1.3 Ancillary Rewards ..................................... 10
    2.1.4 Practical Rewards ..................................... 10
    2.1.5 Difficulty ............................................. 10
  2.2 Types of Game Progression ............................... 11
    2.2.1 In-game Progression .................................... 11
    2.2.2 Player Progression ..................................... 11
    2.2.3 Meta-game Progression ................................ 11
    2.2.4 Player Focused Progression vs Gameplay Focused Progression ................................ 12
  2.3 Related Work ................................................. 12
    2.3.1 My Army: A Game for Social Networks ................ 12
    2.3.2 Ancient and Medieval Battle Simulator ................ 13
  2.4 Relevant examples in the gaming industry - Single Player Focused Games ......................... 13
    2.4.1 Meta-game Progression in XCom2 ...................... 13
    2.4.2 The Total War Genre .................................... 15
  2.5 Relevant examples in the gaming industry - Multiplayer Focused Games ......................... 17
    2.5.1 Total War Arena ....................................... 17
      2.5.1.A The Core Loop ..................................... 17
2.5.1.B The Secondary Loop .................................................. 19
2.5.2 World of Tanks ............................................................. 19
  2.5.2.A The Core loop ......................................................... 19
  2.5.2.B The Secondary loop .................................................. 20
  2.5.2.C Gameplay Enhancement from Persistent Elements ............. 21
2.5.3 Phasmophobia ............................................................ 22
  2.5.3.A The Core Gameplay .................................................. 22
  2.5.3.B Item and Resource Management .................................. 23
  2.5.3.C Game difficulty and how it affects progression .................. 24
  2.5.3.D Game Success ......................................................... 24

3 Mount and Blade II: Bannerlord .............................................. 25
  3.1 Description ..................................................................... 27
    3.1.1 Single Player ............................................................ 27
      3.1.1.A Upgrading and Acquiring Troops ................................. 27
      3.1.1.B Player Character progression ..................................... 28
      3.1.1.C Companions ......................................................... 28
      3.1.1.D Battling ............................................................. 29
      3.1.1.E RPG Elements ..................................................... 29
      3.1.1.F The Clan ........................................................... 30
      3.1.1.G Conclusion ........................................................ 30
    3.1.2 Multiplayer Mode ....................................................... 30

  3.2 Architecture and Structure ................................................. 31
    3.2.1 Interfaces and Other XML Structures .............................. 31
    3.2.2 Important DLLs for the Thesis ....................................... 32

  3.3 Modding ....................................................................... 32
    3.3.1 Project Setup ............................................................ 33
    3.3.2 Harmony .................................................................. 33
    3.3.3 Other Modding Techniques .......................................... 33
    3.3.4 Limitations ............................................................. 34

  3.4 Relevant examples in Bannerlord Multiplayer Modding ............. 34
    3.4.1 Bannerlord Online ...................................................... 34
    3.4.2 Bannerlord Co-Op ....................................................... 35
    3.4.3 Conclusion ............................................................. 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5.2 Match Making Selection</td>
<td>63</td>
</tr>
<tr>
<td>5.5.3 Post-Battle</td>
<td>63</td>
</tr>
<tr>
<td>5.6 Formations and Battle Plan</td>
<td>63</td>
</tr>
<tr>
<td>5.7 Game Balance Changes</td>
<td>64</td>
</tr>
<tr>
<td>5.7.1 Upgrade Cost Being Too Cheap</td>
<td>64</td>
</tr>
<tr>
<td>5.7.2 Differentiating Noble Troops from Regular ones</td>
<td>64</td>
</tr>
<tr>
<td>5.7.3 Elo balancing</td>
<td>64</td>
</tr>
<tr>
<td>5.7.4 Unaddressed Balance Issues</td>
<td>65</td>
</tr>
<tr>
<td>5.8 System Limitations</td>
<td>66</td>
</tr>
<tr>
<td>6 Tests and Feedback</td>
<td>67</td>
</tr>
<tr>
<td>6.1 Mod Release</td>
<td>69</td>
</tr>
<tr>
<td>6.1.1 General Lords’ Page</td>
<td>69</td>
</tr>
<tr>
<td>6.2 Dissemination and Community</td>
<td>70</td>
</tr>
<tr>
<td>6.3 Mod Reception</td>
<td>71</td>
</tr>
<tr>
<td>6.4 Automatic Data Collection</td>
<td>72</td>
</tr>
<tr>
<td>6.4.1 Population Demographics</td>
<td>72</td>
</tr>
<tr>
<td>6.4.2 High Engagement Players</td>
<td>76</td>
</tr>
<tr>
<td>6.5 Open Survey</td>
<td>80</td>
</tr>
<tr>
<td>6.6 Controlled User Tests</td>
<td>82</td>
</tr>
<tr>
<td>6.6.1 The Controlled Tests</td>
<td>82</td>
</tr>
<tr>
<td>6.6.2 The Survey</td>
<td>83</td>
</tr>
<tr>
<td>6.7 Conclusion</td>
<td>85</td>
</tr>
<tr>
<td>7 Conclusions and Future Work</td>
<td>87</td>
</tr>
<tr>
<td>7.1 Conclusions</td>
<td>89</td>
</tr>
<tr>
<td>7.2 System Limitations and Future Work</td>
<td>90</td>
</tr>
<tr>
<td>Bibliography</td>
<td>91</td>
</tr>
<tr>
<td>A Descriptions</td>
<td>95</td>
</tr>
<tr>
<td>A.1 Battle Plans</td>
<td>95</td>
</tr>
<tr>
<td>A.1.1 Phases</td>
<td>95</td>
</tr>
<tr>
<td>A.1.2 Orders</td>
<td>96</td>
</tr>
<tr>
<td>A.2 Mod Description</td>
<td>96</td>
</tr>
<tr>
<td>B Figures</td>
<td>99</td>
</tr>
<tr>
<td>C Open Testing Player Survey</td>
<td>107</td>
</tr>
</tbody>
</table>
D User Test and Survey

D.1 User Test Guide
  D.1.1 General Lords User Test and Consent Form
  D.1.2 Consent Form
  D.1.3 General Lords User Test Guide

D.2 User Test Survey
List of Figures

2.1 Xcom2: Example of battle ......................................................... 14
2.2 Xcom: Motherbase ................................................................. 14
2.3 Xcom: Troop Management ........................................................ 15
2.4 Empire Total War: Example of empire battle ................................. 16
2.5 Rome II Total War: Example of roman battle ................................. 16
2.6 Total War Arena: Example of a battle ........................................ 18
2.7 Total War Arena: Example of troop tech tree ................................ 18
2.8 World of Tanks: Example of a tank battle .................................... 20
2.9 World of Tanks: Example of a tank research tree ............................ 20
2.10 Phasmophobia: Example of gameplay ........................................ 22
2.11 Phasmophobia: Game Evidence and Journal ................................. 23
2.12 Phasmophobia: Buying newer items ......................................... 23
2.13 Phasmophobia: Inventory ....................................................... 24
3.1 Bannerlord: Example of a troop tree ........................................ 27
3.2 Bannerlord: Example of inventory ............................................. 28
3.3 Bannerlord: Example of a Bannerlord battle ................................. 29
3.4 Bannerlord: Example of an army gathering .................................. 30
3.5 Bannerlord Online: Players in a town ........................................ 34
3.6 Bannerlord Co-Op: Two clients connecting to a server .................... 35
4.1 Relation between the in-game loop and the meta-game loop .............. 39
4.2 General Lords: Main Screen .................................................... 39
4.3 General Lords: Example of battle in General Lords ....................... 41
4.4 General Lords: Recruitment Interface ....................................... 44
4.5 General Lords: Character Interface .......................................... 48
4.6 General Lords: Match History Interface ..................................... 48
B.10 Discord’s feedback 6 to the mod ........................................... 101
B.11 Nexus Mods’ feedback 1 to the mod ..................................... 101
B.12 Nexus Mods’ feedback 2 to the mod ..................................... 102
B.13 General Lords trending ......................................................... 102
B.14 General Lords most downloaded ........................................... 102
B.15 Online Survey Usability Likert Answers Distribution ............... 103
B.16 Online Survey Enjoyment Likert Answers Distribution ............. 103
B.17 Online Survey Boxplots ......................................................... 104
B.18 Control Test Usability Likert Answers Distribution .................... 104
B.19 Control Test Enjoyment Likert Answers Distribution ................. 105
B.20 Control Test Boxplots .......................................................... 105
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMO</td>
<td>Massive Multiplayer Online</td>
</tr>
<tr>
<td>XP</td>
<td>Experience</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>DLL</td>
<td>Dynamic-Link Library</td>
</tr>
<tr>
<td>NPC</td>
<td>Non-Player Character</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>CO-OP</td>
<td>Cooperative</td>
</tr>
<tr>
<td>RPG</td>
<td>Role-Playing Game</td>
</tr>
<tr>
<td>MMORPG</td>
<td>Massive Multiplayer Online Role-Playing Game</td>
</tr>
<tr>
<td>PVP</td>
<td>Player versus Player</td>
</tr>
<tr>
<td>HP</td>
<td>Health Points</td>
</tr>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Introduction

Contents

1.1 Motivation ........................................ 3
1.2 The Problem ...................................... 4
1.3 Goals .............................................. 5
1.4 Results ............................................ 5
1.5 Outline ............................................ 5
Persistent elements are a common approach to most games to add a sense of constant progression outside of the main game or main loop. Given its rising popularity we will first define on a larger scope what progression could mean on a broader spectrum in the video-game industry. Progression in its raw definition means “the process of developing gradually from one stage or state to another”\(^1\). In the context of a game it could represent everything that suffers a change over a period of time (or unit of time period represented in game such as a turn).

For example, progression in chess is the change of state between each player’s turn over time. Assuming the players are playing to win, there is a finite amount of states, each successive state slowly culminating into the final state in which a player has won. This is an example of in-game progression, meaning the progression begins and ends in one iteration of the game (one playthrough), each playthrough is independent (losing or winning the previous match will not affect the next) and each playthrough has its own progression (each match is likely different from the others). However, it would be unfair to say there is no progression outside of each match, since over the course of each match (considering each one as our time unit) each player is learning and adapting to the game and to the other player so, in a sense, chess has also a form of “persistent progression” even though not directly created by the game itself.

However, we should take into account that the chess rating system \([4]\), although not a form of gameplay progression, is a form of progression for the player, or rather a metric for the player’s progression and skill. Chess rating tries to classify a player’s skill in the game compared to the other players, which means that when a less rated player (for example 1000) plays versus a more rated player (for example 1500) and wins the game, it could mean the player is slightly underrated and therefore gets more points than usual for a win.

This “persistent progression” is often denominated meta-game progression, and represents the progression that carries over from each playthrough (usually with persistent elements, such as unlockable perks, new items or new game mechanics). A game genre that commonly uses meta-game progression are rogue-like games, like Binding of Isaac \([22]\) or Dead Cells \([24]\), where each time the player plays a game round he could unlock new features that will be available on the following playthroughs.

### 1.1 Motivation

A lot of multiplayer games rely on the same kind of challenges repeated over and over, and we believe there are ways to improve a multiplayer experience that help make this cycle more enjoyable and less repetitive. We want to test if adding a good progression system into the multiplayer experience will improve its overall experience and improve its overall longevity. This will prove to be a challenge, as

\(^1\)Oxford Dictionaries, [https://www.oxfordlearnersdictionaries.com/us/definition/english/progression](https://www.oxfordlearnersdictionaries.com/us/definition/english/progression)
adding a progression system can often lead to the feeling of being overcompensated and meaningless, in the sense that the players feel it brings nothing to the table and shouldn’t even be implemented.

So we want to evaluate whether adding such a system to a multiplayer experience would attract players to play and if such players would enjoy the game mode. We had to decide where and how such a system would be implemented and Mount & Blade II: Bannerlord stood out as a potential candidate for such concept. The game has a fairly strong and developed progression system in the single-player campaign, a game mode of the game, and we feel this system could be used and adapted to complement the game’s multiplayer experience to create a new and fun game mode that players will enjoy and want to play more.

1.2 The Problem

At the moment of writing this thesis, October 27th, 2021, Bannerlord hasn’t got any sort of similar game mode that combines persistent elements to a multiplayer experience. The current implemented system (among other game modes) is a team deathmatch game in which six players battle another six. Each player can select a type of troop squad which he can then command - infantry, ranged units or cavalry. The goal of the battle is to occupy specified tactical points, and whoever owns most points will drain away the enemy’s score; when a team’s score reaches zero the battle is over and the team with the remaining score wins the battle. Battles last everywhere between 15 and 45 minutes.

Whilst the game mode is fun, each battle is a unique event and has no further impact outside of the battle itself. We feel that the experience could be improved if the player had something to do outside of the battle that would impact future battles in one way or another.

We should keep in mind that adding means of progression does not guarantee a more enjoyable progression and some games, for example Among Us [17] that was one of the most popular games in 2020, don’t require any kind of progression to remain popular and in general be a lot of fun with the game mode itself. In contrast, we have Phasmophobia [27], that has an in-depth persistent inventory system that can be lost with a death in-game. Is a progression system required for a more complete game experience?

With this thesis we aim to evaluate if the new multiplayer mode created a new, fun and enjoyable experience overall and how adding persistent elements contributed to it. Currently there is no lasting relation by the player to the troops he commands, given the temporary characteristics of the game mode. Is a persistent and evolving army a fun mechanic for the game?
1.3 Goals

The main goal is to understand how a meta-game progression system could be designed and implemented into a game such as *Bannerlord* and the impact on the overall experience of the game. We will also try to make a link between common meta-game progression systems in single-player and how they could be implemented in a multiplayer game.

As mentioned before, we want to test the concept in the context of the game *Bannerlord* and implement a proof of concept capable of simulating a progression system that achieves persistent element progression in a multiplayer experience. The *Bannerlord* single-player system already implements a very realistic personal army experience, which means there are plenty of concepts that can be re-used and added on the multiplayer progression system.

We adapted many of these features to our proof of concept with the goal of creating a familiar experience to the player in a new environment; the player should experience the positive aspects in the single-player campaign when playing the multiplayer game mode.

1.4 Results

We developed an asynchronous mod for the game *Bannerlord* that allows the player to challenge multiple player armies whilst maintaining a permanent army that evolves over time. The mod was then released in Nexus Mods, a popular mod distributing platform, on the 26th of August.

Along side the release, a questionnaire was distributed for players to evaluate if they enjoyed the mod created and asking them to give feedback on the overall experience. In addition, controlled tests were performed to understand how the mod would fare with newer people with no real previous experience in the game.

The mod had good ratings in both surveys; it did however fail to captivate the majority of testers for longer than a day. Those that did try for more than a day, usually enjoyed the mod rather a lot and some even invested over 30 hours during the span of a month.

1.5 Outline

In this document, in the *Related Work* chapter, we will first analyse persistent progression and the current state of the art in regards to meta progression and how it is applied to current examples of games.

In the next chapter, *Mount and Blade II: Bannerlord*, we will give a brief overview of the game chosen to develop the mod and in addition we will give light to the overall modding experience in the
game and the current competition to our mod. The next chapter, the **Conceptual Model**, will describe the mod on a high level and how its systems are designed.

Then we will describe the steps taken in the implementation and the architecture of the mod in the **Implementation** chapter, along with the server required for the multiplayer. We will also analyse the balance changes made throughout the testing span.

The **Tests and Feedback** chapter will go over the mod’s release stage, results and feedback on the mod.

And to conclude this document we will give our final conclusions in the chapter **Conclusions** on how the overall process went and what are the future steps in regards of the mod.
2 Related Work

Contents

2.1 Game Progression Theory ................................................. 9
2.2 Types of Game Progression ............................................. 11
2.3 Related Work .......................................................... 12
2.4 Relevant examples in the gaming industry - Single Player Focused Games ........... 13
2.5 Relevant examples in the gaming industry - Multiplayer Focused Games ........... 17
In this chapter we will further develop on the state of the art for meta progression and persistent elements used to achieve said progression, as well as addressing its possible effects. To achieve this, we will analyse documents and blogs on progression and the current implementations in modern relevant games. We will start by analysing in-depth the concept of game progression and how it leads up to meta progression.

2.1 Game Progression Theory

As explained previously, progression is the continuation and advancement of a state towards another, usually, more advanced state. This translates fairly well to game design and gameplay.

There are other ways of interpreting progression in games, in the book “Game Mechanics Advanced Game Design” [1], where Adams contrasts emergence with progression as the difference between a more “sandbox” style or low rule based game with a more strict and linear narrative game. The concept is further cemented in the paper “Assessing emergence and progression in games” [8], by Maranhão, Franco and Maia. This is however, not the interpretation of progression we will approach in this thesis.

In the article “Gameplay Design Fundamentals: Gameplay Progression” [7] the author Mike Lopez describes Gameplay Progression with five key elements: Game Mechanics, Experience Duration, Ancillary Rewards, Practical Rewards (gameplay relevant) and Difficulty.

2.1.1 Game Mechanics

Game mechanics are the controls and interactions in a game. These controls are usually tied to weapons, abilities, new forms of movement, powers and other related systems. Progression integrates with game mechanics in the sense that it helps structure when these mechanics are available and shown to the player. The most common methods of progression are Gated Access and Directed Gameplay

• **Gated Access** In gated access the more advanced mechanics are gradually unlocked throughout the game. Usually unlocked by leveling up or reaching a certain level.

• **Directed Gameplay** In directed gameplay most mechanics are unlocked from the start but the levels are built in such a way as to direct the player to each feature over time. The first levels require only the most basic ones to finish, this way the newer players aren’t overwhelmed by too many features and more veteran players have easy access to more advanced features and can clear faster the early levels.
2.1.2 Experience Duration

The length of the experience in a game doesn’t necessarily need to increase over time for a game to be successful. However, some games benefit from longer playthroughs, for example, racing games unlocking larger races that take more time to finish. This complements well with added mechanics (new mechanics help the player finish levels faster therefore the levels can be bigger to balance it out).

2.1.3 Ancillary Rewards

These are rewards that, even though have no impact in-game, give a player a sense of gratification when achieved and increase replayability, like mentioned by Mike Lopez [7]. A simple example for the race game would be medals for performance in a race, if the player meets the required threshold he will get a medal based on that same threshold, therefore a more perfectionist player could play the same race multiple times to get the highest medal.

There are also ancillary rewards that are aimed at measuring the overall player progression. One of such examples are rankings, an abstract measure with the goal of classifying players in comparison with each other in terms of game performance. Like mentioned before, the elo rating system of chess [4], is a form of ranking players in terms of their capabilities compared to others.

Besides being a good ego boost for players, knowing they are among the best, it also has practical applications, for example tournament management (ensuring all players in a tournament are around the same skill level).

2.1.4 Practical Rewards

Practical rewards link with gameplay mechanics in the sense that often is through practical rewards that the player improves his abilities and items and how he unlocks new modes and/or characters. Practical rewards differ from ancillary rewards in the sense that they affect gameplay where ancillary rewards do not.

2.1.5 Difficulty

Lastly, the progression of difficulty of a game greatly affects how well it’s received. It’s really hard to balance out the progression of difficulty in a game, should the game be too easy it won’t keep players engaged, on the other hand if the game has a learning curvature too steep the newer players will get frustrated and stop playing the game.

Nowadays most games balance out in an exponential curvature, the initial game is easy enough to help newer players and the later stages of the game being more difficult should motivate the more
veteran players to steam roll the early stages to get to the harder parts.

2.2 Types of Game Progression

We can subdivide the concept of progression into three main types: the in-game progression, the player progression and the external game progression, usually called meta-game progression.

2.2.1 In-game Progression

The in-game progression, or the standard progression (the change of state in the main loop of the game), is where most of the key features mentioned before come to fruition and are explorable by the player. The in-game progression is everything related to the overall progression in the main loop. The main loop represents the core gameplay of a game, for example a battle in a strategy game or a race in a racing game.

Taking in the example of the racing game, every time a player starts a race he is partaking in the main loop. So the steps the player takes in the race to finish it are the overall progression of gameplay, this is where the previously mentioned features take place.

It's also in the in-game progression how the flow of the game is defined. In the book “The Art of Game Design - A Book of Lenses” [13], Jessie Schell cites an interesting critique he got when he was young: “You have some good stuff in this act, but the progression isn’t quite right.”. He then explains that even thought the game is solid, there needs to be a proper sequence of exposition, be it game story or be it game mechanics.

2.2.2 Player Progression

Another type of progression is the player’s personal progression, this being every time the player has an experience or is forced to adapt to a challenging moment that appears in the game. Usually, if the player’s skill does not progress in a game, he will in the end fail over and over, the main reason being the previously mentioned feature “Difficulty”, since the difficulty of a game will scale over time, and the player needs to adapt to this change.

2.2.3 Meta-game Progression

Meta progression in short is represented in this form, as explained in the article “The Procession of Progression in Game Design” [3] by the author Josh Bycer: “A game featuring a main game system, and a secondary one that adds permanence between play sessions”. In the article the author also mentions the other forms of progression and how they could tangle between meta progression.
Meta-game progression is represented by the features that carry over from each iteration of the main loop, taking into example again a racing game, we can say that the car could be a permanent feature that the player brings each race. The progression usually comes from a secondary loop (could be a waiting hub or area the player stays during the in-between iterations of the main loop) where the player manages and upgrades the permanent assets that are used in the main loop.

2.2.4 Player Focused Progression vs Gameplay Focused Progression

As mentioned in the article by Bycer [3], games can juggle between player focused progression and (internal or external) gameplay focused progression. If we take in consideration a racing game again as an example, there is a balance between the player’s skill and the available mechanics. At an early stage the player needs to have skill in racing games to finish first, but as the game progresses it is expected that the player’s opponents become better and have access to better cars. This concludes in the player having to upgrade his car regardless of his skills; there is a balance between the player getting better at the game and acquiring better gear.

Therefore regardless if the player has good cars he will not succeed if his driving skills are poor and, on the other hand, even if the player is really skilled he will still need to engage and develop his car and gear to even have a chance at succeeding. This balance ensures the game doesn’t get too obvious for the player and ensures even really skilled players can still have a good experience with the game.

Games can also specifically focus on one or the other type of progression as their core gameplay. Games can focus solely on gameplay focused progression, without requiring any real form of player focused progression. One of such examples are time sink games, like the popular mobile game AFK Arena [16], were players perform a various set of gameplay actions that require little to no learning curve to adapt. In contrast players can have a rather steep player focus learning curve, like Dark Souls [23], were players must invest multiple hours in the same feature to perfect their craft.

2.3 Related Work

There has been previous work done related to progression and meta-game progression, as well as work relevant to our expected approach to the Bannerlord’s multiplayer.

2.3.1 My Army: A Game for Social Networks

In the thesis “My Army: A Game for Social Networks” [10] by João Morais the concept of a persistent personal army is analysed in detail. Even though on a simpler game, João Morais explains how to create a good battle cycle: the players battle, invest the rewards in the army, prepares tactics to battle new
players. This cycle creates the balance between battle and meta-game progression needed to keep the game engaging.

2.3.2 Ancient and Medieval Battle Simulator

The thesis “Ancient and Medieval Battle Simulator” [15] by Pedro Vaz gives an approach to the concept of battle preparation. Most games have either no battle at all or no real preparation and set-up of a battle. Historically this was not the case, the positioning of an army in the beginning of a battle could often determine the course of the battle, as Pedro Vaz gives the example of the battle of Cannae. Bannerlord too has no battle preparation, and this could be a concept that can be explored to further improve the game’s multiplayer experience.

2.4 Relevant examples in the gaming industry - Single Player Focused Games

To further analyse progression and, more importantly, meta-game progression we will analyse a few relevant games. We will analyse first how meta progression is usually implemented in single-player based games and then how they translate to multiplayer and how multiplayer could benefit from common features of progression in single-player games.

2.4.1 Meta-game Progression in XCom2

In the article by Bycer [3], it’s explained how meta progression is executed in the game XCom2, which is actually really close to what we pretend to achieve. The main game loop of XCom2 is each battle taken between the player’s squad and a battalion of enemies (groups of enemy squads spread throughout the map); however each battle can have different objectives and end goals for success such as: search and destroy (destroying a specific statue or planting a bomb in a facility), hostage rescue (saving a village from an attack or escorting a VIP to the extraction point) and elimination (it is the most common game mode and all the player has to do is eliminate all enemy units). If this was the only game experience, the game would have only what we can describe as in-game progression (each game mode has however a different form of mission progression). This progression being: the way and tactics the player has to employ to succeed in each mission. And like Bannerlord, if this was all the experience the game had, it would have a certain linearity. The player would bring his squad and do missions over and over again until he eventually got bored.

\[1\] Taken from https://www.youtube.com/watch?v=mwL_3Nf7o8g&ab_channel=%C6%92ru77y
This is where the *XCom2*’s secondary loop comes to change the flow of the game. The secondary loop takes place between each battle, has great effect on how future battles will turn out and is greatly affected by previous ones. Essentially, after each battle the player receives some resources and experiences for his squad troops, with the disadvantage that should a squad mate die in battle, he/she will be lost permanently.

As it can be inferred, losing a squad member will greatly affect future battles, and not only that but future battles will get considerably harder; this means that even though the current squad can win the current battle, it might not be able to win the following one. This is where the secondary loop comes in, here the player should manage the acquired resources (from battle and over time passive gains) in

---

1. Taken from https://herringsfishbait.com/2016/04/22/xcom-2-legendary-ironman-guide-tips/
2. Taken from https://www.youtube.com/watch?v=Tutk2ylofYk&ab_channel=IGN
order to strengthen the current squad (the survivors of the previous squad and future recruits used to replace the lost members) so the squad is up to par with the enemy.

We should keep in mind that these loops are open to interpretation and that some players could consider the management loop as the focus of the game rather than a secondary loop. In this thesis however, the interpretation of the meta-game loop serves as an extension of the in-game loop and therefore is secondary to it.

Besides meta-game progression helping fix many problems with game balancing (as it can be observed in the previously mentioned game where a player could only fight harder battles when he upgraded properly his troops), it also ensures the game remains fresh and each experience of the core game-play loop is different and constantly changing. Citing the article by Bycer [3]: “Even though the player may be playing the same maps or game modes over and over again as before, they’re now using new weapons and abilities to break up the monotony.” This is precisely our goal for the Bannerlord’s multiplayer experience. By integrating a concept like meta progression we ensure each battle has a fresh feel since the player’s troops are different, personalised and are affected by the success or failure of the battle.

2.4.2 The Total War Genre

The Total War Saga is a really good example of single-player meta-game progression. Developed by Creative Assembly, these games have a fairly unique approach to strategy games.

All games feature a battle simulator as a core gameplay loop, and although the feel of the battles change from game to game, the core is always the same. Two armies (or more) fight a battle until the last army stands. The strategies of the battles may vary between each game: in Total War: Empire [30] the battles follow the line formation used in the Imperialistic era and cannons are key for victories, where
as in *Total War: Rome II* [31] the battles have a slower pace and follow the historical shield formation and Roman testudos.

![Example of empire battle](image)

**Figure 2.4: Example of empire battle**

![Example of roman battle](image)

**Figure 2.5: Example of roman battle**

Even though the battles can feel fairly different by themselves, the identity of these games also comes from the map mode and country building feature which functions as the secondary loop of this type of games.

**Total War game progression:** The standard loop of a Total War game is starting out in a relevant country to the era the game plays in (for example Rome for *Total War: Rome II* or Great Britain for

---

4 Taken from [http://etw.heavengames.com/info/demo-review/](http://etw.heavengames.com/info/demo-review/)
5 Taken from [https://www.honga.net/totalwar/rome2/unit.php?l=en&v=rome2&f=rom_pontus&u=Gre_Bronze_Shield_Pike](https://www.honga.net/totalwar/rome2/unit.php?l=en&v=rome2&f=rom_pontus&u=Gre_Bronze_Shield_Pike)
Total War: Empire) in a semi-sandbox style where the player can build and lead the country to his own desires.

The early phases are riddled with challenges since usually the players start really under-powered in all aspects: small armies, low income, poor quality troops, etc... The game then focuses on spending the early turns in training armies, improving economy and preparing for war with the nearby countries.

Over time, as the player conquers newer territories and improves the current ones, builds better armies through resources and technology, the player will eventually come to be a powerhouse and will reach the late stage of the game where the player is far too strong to nearby powers and slowly conquers all of them.

Player relation to Progression: In these games, players cannot ignore progression. Armies deplete fast and have to be retrained, they require money which requires production and industry. In short, the player needs to focus on improving his army should he have any desire to succeed in battle. This comes from the fact that Artificial Intelligence (AI) controlled countries will not stay idle and they too will spend their turns improving and recruiting bigger and better armies.

In this case, the fact that the enemy progresses constantly is a big motivator for the player to do the same. This concept can be bridged to the multiplayer scenario since human players will want to improve their armies to fight versus other human players.

2.5 Relevant examples in the gaming industry - Multiplayer Focused Games

2.5.1 Total War Arena

Total War Arena [34] is a strategy game developed by Creative Assembly, the same creators of the Total War Saga. It’s a free-to-play title and focuses on the online multiplayer experience. It consists of a strategy team deathmatch multiplayer with persistent elements between each battle. To further analyse the game let’s specify that the game has a core gameplay loop and a secondary gameplay loop that intertwines with the first one.

2.5.1.A The Core Loop

The core loop is the deathmatch in the “arena” (the battleground where the teams’ armies battle each other). In essence, the deathmatch consists of a showdown between two teams of 10 players each, where each player controls three squads of troops. These squads are around 100-200 soldiers each and can vary between different types (infantry, ranged units, cavalry and siege weapons). The goal of

6 Taken from https://worldoftanks.eu/en/news/general-news/total-war-arena/
the game is to capture key points in the map to drain away the team’s score (the team which the score reaches zero loses) or be the last team standing (the last team with alive troops). Besides the player’s squads, each player can choose a general from a pool of generals (it’s possible for players to bring the same generals as their teammates).

This experience by itself is fairly complete, since we can envision a complete multiplayer experience with just these characteristics, each round the player selects one general and 3 squads from the available types and plays a team deathmatch over and over again. In a sense, it’s precisely the experience Bannerlord’s multiplayer currently gives.
2.5.1.B The Secondary Loop

There is however a squad management moment between each game, and, although it is not required by the player to engage in these mechanics, like in the example in XCom2, the player will eventually fall behind to the players that do engage in the secondary loop. So rather than assigning random squads and random generals, the player is given squads that progress over time (the current army is not a one time instance that is reset but a constant element that is used each battle, meaning previous battles will affect the state of the army and the said state will affect the future battles).

So outside of each battle there is a research loop where players use acquired resources to level up his squads, equip them with better weapons and armor and unlock new unit types using squad experience; there is a management loop where players define loadouts (combinations of squads that players want to bring tied to the already unlocked troops). The general has also a progression tree where players unlock new abilities and upgrade current ones to use in battle (these abilities are often passive buffs to the troops or active commands that players can execute in-game).

However, the player can’t spend an infinite amount of time micro managing his squad without battling frequently; to balance out how much time the player spends managing his squad’s stats, the player needs to engage frequently in battles to gather the resources needed for upgrades (coins and squad experience). This is where the link and balance comes from between both moments of the game, the player needs to actively fight battles if he wants to upgrade and progress his troops and needs to spend time upgrading and managing his troops should he desire to keep progressing and have stronger units in battle.

2.5.2 World of Tanks

World of Tanks [32] is a Massive Multiplayer Online (MMO) developed by Wargaming, and the game was launched in August 12th, 2010. It’s a free-to-play game with in-game purchases, and much like the previously mentioned Arena, the game focuses on a team deathmatch multiplayer with persistent elements between the battles. Also, like the previous game, it features a core loop and a secondary loop.

2.5.2.A The Core loop

The core loop are tank battles between two teams of fifteen players each. The major difference to the previous game is that in this one the player controls only a tank in third person view; there are many types of tanks (heavy, medium, light and siege tanks are the main classes but there are others) that offer different characteristics and require different play-styles to have success in the game.

7Taken from https://totalwar.fandom.com/wiki/Talk:Total_War:_Arena
8Taken from https://worldoftanks.eu/en/game/
A note worthy feature of the core loop is its in-depth damage system that simulates realistic damage to what would happen to the said tank in real life. The game takes into account bullet speed, angle of hit, armor penetration based on angle and can even rebounce the bullet if the angle is too big for a direct hit. This system will later be mentioned with a closer relation to the end goal of the thesis.

2.5.2.B The Secondary loop

The secondary loop of World of Tanks focuses on the research, development and upgrading of tanks and acquisition of new different models of tanks (all between 1930-1960s era tanks). There are many different aspects to tank management [21] so we will begin by analysing how an individual tank progresses (before unlocking other tanks).

Taken from https://na.wargaming.net/support/en/products/wotb/article/15407/
A tank has five major modules (the turret, the gun, the engine, the tracks and the radio) and other minor slots which have several options with passive effects for the tank in question. The way a tank is upgraded is by unlocking new versions of one of each module with Experience (XP), then buying the module using the common currency “silver”, so over time a single type of tank can be upgraded to be several times stronger than its most basic version. The research is done by a research tree and some modules are blocked further ahead which means the player sometimes has to research unwanted (or less desired) modules to reach the more valuable ones. In this way the balance is ensured and guarantees the tank doesn’t ramp up too much, making the other upgrades almost meaningless. There is also another soft block which is the tank’s weight, the player needs to upgrade the tracks module to have a bigger capacity and put heavier modules (usually better than the lighter counter-parts).

Some modules also have compatibility with other tanks, which means that sometimes when a player gets the next tank in line he can skip past some modules of the research tree (since he already had them researched on a previous tank).

Now comes the other important aspect to the persistent based meta-progression of this game, which is more tanks and tier upgrades. Tier upgrades are improved versions of a type of tank, we can conclude from the previous explanation that a tank can have a tier and each tier version can have several combinations of modules. There are also several available different tanks which the player can unlock (the player needs vehicle slots to have different tanks) therefore, following the previous conclusion, a player can have more than one tank that has a tier which can be upgraded once newer tiers are unlocked and each tier has an individual research tree that unlocks new modules (this tree can intertwine with other tank tiers and even other tanks).

There are also many other features that rely on persistent based elements but we will now talk about a really relevant one, which is the ammo type loadouts the player can bring into battle. The player can define how much ammo and of which type it is.

2.5.2.C Gameplay Enhancement from Persistent Elements

Like mentioned before the damage system is really sophisticated, and is further enhanced by the ammo type which the player fires; there is armor penetration, heat rounds and other less popular types. The key point to take is that the system is further improved since different rounds have different reactions on impact and given the limitation of the ammo (since the player has to buy them previously on the secondary loop) the player has to decide which is the best use of each type so he does not waste his good rounds on not so efficient shots. This gives a bigger strategy in-game that stems from an outside influence which is how much ammo the player has.
2.5.3 Phasmophobia

*Phasmophobia* [29] is a Cooperative (CO-OP) multiplayer game created, developed and published by *Kinetic Games* in September 2020. The game’s experience focuses on horror gameplay, where one to four players gather at a sightings scene (a haunted house or school and others) and use ghost hunter’s techniques to discover what type of ghost they are dealing with.

2.5.3.A The Core Gameplay

The players start on a van decked out with ghost hunting props and the players need to use them to get clues on what type of ghost the current site has. Each ghost type has 3 respective clues out of 7 total, and there are 12 ghost types so there are a lot of different combinations of clues that lead to different ghost types.

The players can’t simply bring all the equipment to the house at once, they have three item slots and usually (but not necessarily) one of them will be used up for the flashlight; the flashlights are crucial since the game is rather dark with almost no illumination and also warn the player when the ghost is attacking by flashing intermittently. Then the player usually brings the starting items like the thermometer to pin point the ghost location (temperatures under ten mark the exact position the ghost is located) and then use all the specific items for each clue to gather them all.

Eventually with the players’ activities around the ghost, it will decide to attack (more frequently the less the sanity the players have) to which the only solution is hide and wait out the attack.
2.5.3.B Item and Resource Management

There is a twist to the gameplay, the items are perishable [28], each time the player dies to the ghost the player loses his items and needs to buy new ones in the game lobby. To buy them the players need to successively complete contracts in order to earn money to buy them.

A new player will start the game with only a basic set of items that allow the player to get only a few of the clues (which means either he guesses the rest or focuses on the side objectives which also earn the player money). After a while the player will earn enough money to buy the full set of items which at
this point the player will only have to save money for an eventual death and item loss.

### 2.5.3.C Game difficulty and how it affects progression

There is also a leveling up system, as in each successful game the players play they earn XP and eventually level up. There are several items locked out with level requirements, for example the sound sensor is only available for purchase at level 8.

At level 10 the player is allowed to choose a higher difficulty (intermediate) and at 15 he gets the hardest difficulty (professional). The key difference from these difficulties besides a harder game is the players earn more money and more XP than the previous difficulties and get less money returned from lost items (player’s death). The players get 50 percent of the money spent back at amateur level, 25 percent at intermediate and 0 percent at professional.

### 2.5.3.D Game Success

The common consensus [14] [33] is that the massive success of this game (it was the top selling game on steam at launch and the sixth most watched game on twitch) is given to its fantastic job in implementing a scary game. However, we should take in to account all the underlying reasons for being such a great game and one of them is definitely its inventory system. Although not massively complex it’s fairly crucial to the experience and the chance to lose its progress conveys an even more scary experience.

This comes from the fact that the fear from the ghost doesn’t come only from its scary features alone but also comes from the consequences of death, making the player be more impacted with each death.
Mount and Blade II: Bannerlord

Contents

3.1 Description ......................................................... 27
3.2 Architecture and Structure ........................................ 31
3.3 Modding ............................................................... 32
3.4 Relevant examples in Bannerlord Multiplayer Modding ............ 34
In this chapter we will give a brief overview of the game *Mount and Blade II: Bannerlord* and its core mechanics as well as the overall modding experience and current competition to our mod’s concept.

### 3.1 Description

*Mount and Blade II: Bannerlord* is an early access game which focuses on early medieval warfare, developed by *TaleWorlds Entertainment*. It’s a strategy action Role-Playing Game (RPG) that envisions recreating the life of a commander in the medieval era. It was launched on March 30th, 2020 and is a prequel to the game *Mount and Blade Warband*, also developed by *TaleWorlds Entertainment*.

#### 3.1.1 Single Player

The single player experience is based around an avatar created by the player which has the ability to command a small party of troops. The core game loop relies on fighting Non-Player Character (NPC) lead parties belonging to factions present in the game. To improve the party’s capabilities, the player recruits new troops, upgrades current ones and levels up his character so he can fight harder and better equipped parties.

#### 3.1.1.A Upgrading and Acquiring Troops

Given the party building focus of the game, the recruiting system is fairly important for the game; the player can recruit troops in two main ways: hiring mercenaries at the tavern or recruiting peasants and recruits in villages and settlements. The next step is upgrading these troops, the upgrade system is

---

1 Taken from [https://www.reddit.com/r/MB2Bannerlord/comments/fwhpqi/mount_blade_2_bannerlord_troop_trees/](https://www.reddit.com/r/MB2Bannerlord/comments/fwhpqi/mount_blade_2_bannerlord_troop_trees/)
based around troop trees (progressive trees where the player unlocks each troop tier following the tree branches). There are many different troop trees (mainly the mercenary tree and each faction’s tree). To upgrade them the troops need to reach a certain level (specific for each troop tier in the tree) by gaining XP from battles.

3.1.1.B Player Character progression

There is also a fairly complex system dedicated to the player’s avatar; besides the cosmetic features (the player can customize the appearance of his avatar) there are a lot of features linked to the player’s character. The inventory system is where the player can see his inventory (food for his party, trade goods, weapons and armor), here the player can also change his character’s items, such as what weapons to bring and his armor, and add a horse and its respective armor.

There is also a leveling up system where the player can level up particular skills linked to in-game actions (sword fighting skills to fight better or athletics to run faster and others). After a certain threshold of each skill the player can select a perk from the available options at that threshold, these perks offer buffs and passives to the player and even the party.

3.1.1.C Companions

The player can also acquire companions which function similar to the player’s character, they also have equipment and armor like the player and also have the same leveling up system with perks and skills. The player can acquire different companions at the taverns they are located at.

A companion can also do side quests for the player and be sent as an emissary to lords along side other available options. The default is having the companion join the player in his party and fight along side him.

---

2 Taken from https://www.pcinvasion.com/mount-blade-ii-bannerlord-how-to-increase-inventory-capacity-and-army-speed/
3.1.1.D Battling

Battles are not predetermined events, a player needs to engage an enemy party or settlement to start a battle. Battle enemies vary from bandits to powerful lords or even kings. As expected, the relative difficulty of an enemy is tied to his/hers role in the world, bandits being the easiest enemy of them all. The overall battles are fought between two teams (each team can have more than one party leader participating) and most of the time the winner is the last standing army. Afterwards the player earns rewards like player XP, party XP, items, gold and renown.

3.1.1.E RPG Elements

Besides the core gameplay of fighting battles and upgrading the party, there are role playing elements such as talking and interacting with NPCs. There are many reasons for the player to interact with NPCs: to start new quests, buy items, receive information (the last known location of key characters, information about the NPC itself, gossip and secrets, etc...) and to partake in political endeavours or romancing/marrying NPCs.

The game has a political system linked to all the factions, and each faction has a hierarchy that the player can join and integrate into, the player then accumulates “influence” which he can use to persuade nobles to support their cause (joining the player’s party to campaign on the map or supporting the player in receiving the next available holding). There is also the possibility for the player to create his own faction and start a new kingdom.

Given the open world sandbox style of game, it allows plenty of play styles such as a trader focused game with caravans and manufactures to make the player wealthy, tournaments that allow the player to gain renown as a skilled fighter and, should the player’s ambitions desire, he can even become ruler of a faction and create his own kingdom.

\[^3\]Taken from [https://www.pcinvasion.com/mount-blade-ii-bannerlord-how-to-increase-inventory-capacity-and-army-speed/](https://www.pcinvasion.com/mount-blade-ii-bannerlord-how-to-increase-inventory-capacity-and-army-speed/)

\[^4\]Taken from [https://www.reddit.com/r/Bannerlord/comments/gg7leq/biggest_battle_youve_fought_in_campaign/](https://www.reddit.com/r/Bannerlord/comments/gg7leq/biggest_battle_youve_fought_in_campaign/)
3.1.1.F The Clan

A new feature Bannerlord has brought in comparison to the previous games is the clan system is that the player has a dynasty which he can evolve when he achieves a certain renown level; the player then unlocks a new clan tier which offers various benefits. This mechanic allows the player to add new parties to the map, usually led by companions or family members, that are a subset of the player’s army. The player can freely gather all his clan parties into a single campaigning army for free.

Each clan tier also dictates how many troops a player can have in his party, how many companions and how many extra parties he can deploy on the campaign map. Along side the roaming parties there are also caravans which function like a party but have a single goal of generating money by trading from town to town.

3.1.1.G Conclusion

Given the complexity of the game it isn’t feasible to cover all of the game in this document, but these are the main features of the single player that will be relevant to the thesis and to give a good overall idea of the single-player. Some of these features such as the army and character progression will be integrated into the main mod which will be described in chapter 4.

3.1.2 Multiplayer Mode

The multiplayer experience currently consists in four game modes: Siege, Team Deathmatch, Skirmish and Captain. The player can select a troop class for his character beforehand and customize his character.

- **Siege**: The siege mode is a 60 versus 60 players game where an attacking team tries to capture a castle from the defending team. The goal of the mode is to capture all the points of the castle
until one of the team's score, referred to as “Morale”, runs out (the team that controls most points drains away the other's score).

- **Team Deathmatch**: Similar to the siege mode, team deathmatch is a massive arena game where players battle in a medieval scenario; the only main difference to the siege mode is that there are no attackers or defenders, it uses the similar point capture of the game and the team that loses is the one that loses all its “morale” or the team with least morale when the time runs out.

- **Skirmish**: Skirmish is really similar to the team deathmatch but is smaller in scale, as it is only between 12 players, 6 vs 6 and they battle in a smaller map.

- **Captain**: This mode is the closest to the original Single-player experience where 2 teams fight a battle of 6 vs 6, where each player controls his own small army. The amount of troops are determined by the strength of the type of troops selected, and the player’s squad type can be specified by the player at the start of the battle and can be one of three types: ranged units, infantry and cavalry.

### 3.2 Architecture and Structure

The game engine was built mainly using C#. We don’t have direct access to the original code, but we can use a software called *dnSpy* which allows us to decompile the Dynamic-Link Library (DLL), which is compiled code, of the game to help us get a grasp of the overall game’s structure and understand how some of its components were built. Besides showing us the decompiled code, it also allows for easy debugging and testing for features, as it allows us to set up break points in the original code. This helps us understand how some features work and were does the code impact the game itself.

These DLLs are mostly contained in the game’s bin folder and there are some auxiliar DLLs that are present in the game’s native modules (add ons to the game created by the game developers that are not necessarily native to the game but are needed for the game’s correct functioning).

#### 3.2.1 Interfaces and Other XML Structures

Not all features are coded in C#, some code is stored in the Extensible Markup Language (XML) format that contains prefabs and templates for certain interfaces, item descriptions, character descriptions and other data driven in-game features.

Creating an interface basically requires three components: the XML file detailing the layout and look of the interface, the C# file which specifies how the overall logic of the interface will work and a framework created by the game’s developers called “the Gauntlet”, which is essentially the link between the two.
if one were to create for example a button for the game, he would first have to create or add this button to an XML file and then specify what this button would do in a C# file and link the two using the Gauntlet.

### 3.2.2 Important DLLs for the Thesis

The game is really vast and contains a lot of DLLs with a lot of code, so we will only overview the most important ones used in the context of this thesis. These DLLs are the following: the Core, the MountAndBlade, the CampaignSystem and the respective ViewModel DLLs.

The **Core** component of the game contains most basic functions and structures of the game. It is crucial for modding compatibility as most data structures are implemented here and are necessary for the correct interaction with the game.

The next crucial component is the **MountAndBlade** DLL which contains more advanced and **Bannerlord** specific features. These features are more focused in the game itself and are needed to easily communicate with the game without having to create them from scratch.

Finally we have the **CampaignSystem**, and it's here that all the major single-player campaign features are implemented and handled, this DLL is crucial to easily adapt and implement the persistent features of the game into the mod. Some examples of these features are the Party System which manages the troops and the upgrade of troops, the Character System that manages the main character and many other persistent elements present in the mod.

As a sub component we have the **ViewModels**, several DLLs that contain most of the logic for the Gauntlet to implement specific User Interface (UI). These are rather important for understanding how interfaces work and how to create new interfaces for modding.

### 3.3 Modding

We will now approach the notion of modding in **Bannerlord** in depth as it is the focus of this thesis. Mod can stand for two words, modification and module, and in this context it stands for both, we will modify the game flow of **Bannerlord** by adding a module to the game.

Modding **Bannerlord**, in essence, consists of creating a project that imports sections of the base code (based on needed features) and calls upon or adds new methods and features to create a new logic for the game. The project is then compiled and added to a folder specifically made for mods in the base game directory. When the game is launched, it will integrate the mod into the game and run its methods as needed.

Given the lack of modding tutorials and resources, since the game is really recent and still in early access, most of the modding had to be learned by reverse engineering how other modders (name commonly used to describe a person modding a game) implemented their mods and how the game
itself implements its features. This is why dnSpy is crucial as it allows us to analyse how most of the code works.

### 3.3.1 Project Setup

The easiest way to setup a new project is by using a tool created by Dealman called “BannerlordMod-Template” [26], which automatically sets up Visual Studio with the configurations and references necessary to work with for modding the base game. It also gives the modder the option to set up Harmony (that will be explained in the next section) compatibility on his project.

When the modder wishes to publish the mod, either for testing or for releasing, he builds the project which will automatically build into the modules’ folder in the game’s folder. For distribution purposes we share this build and whoever downloads it only has to add it to their game’s module folder.

### 3.3.2 Harmony

Harmony [25] is a powerful tool that allows the modder to inject code into the game’s base method. This usually allows to reach otherwise unreachable code (because it is either private or protected) and manually change it to follow the desired results.

As mentioned before, Harmony is really powerful and gives more tools than we necessarily need. Therefore, for simplicity, we will explain the two main tools that were used in this thesis: “Prefix” and “Postfix”. Prefixing allows the modder to inject code prior to the main method’s functioning (it allows to change variables to modify the future behavior of the method or it can outright ignore the rest of the code, effectively completely overriding the method). Postfixing allows the modder to add code at the end of the method (this is mostly used to add extra functionality to an already complete method, but it also allows to override the result returned by the said method).

### 3.3.3 Other Modding Techniques

The modder doesn’t necessarily have to edit the game itself by overriding it with Harmony, some things can be extended and implemented only in the mod’s “workspace”.

For example, we can extend the ViewModel class to create our own interface, and we can add an XML file in the mod’s folder in the game’s module folder. The Gauntlet will then compile this and we will have access to new interfaces created by us in the game.

The game developers (also commonly referred to as devs) also added several overridable methods that allow easy integration in various moments of the game loop. Some examples of these methods are: OnSubModuleLoad, which is executed when the modules are loaded at the launch of the game;
OnGameStart, which allows the modder to add behaviours when the game begins; OnApplicationTick, which is the update function allowing the modder to add recurrent events to the game.

3.3.4 Limitations

With enough time and experience, most if not all features of the game are moddable, either through the resource of Harmony or by the freedom the code itself gives to the modder.

However several systems are rather “hardcoded” (meaning they are not flexible enough to address new situations that weren’t previously envisioned for that system). Some systems rely too much in private methods and read only variables that hinder too much the flexibility of modding deeper systems of the game. Even with Harmony, sometimes a system has too much components and it would take a long time to correctly override them all.

3.4 Relevant examples in Bannerlord Multiplayer Modding

During the development of this thesis two mods were made publicly known and stood out as the first possible multiplayer modding experiences.

Up until this time multiplayer modding was still in question if it was possible given the lack of multiplayer modding tools, these two mods help prove it was possible and we will now analyse a bit of what these mods consist of.

3.4.1 Bannerlord Online

![Figure 3.5: Players in a town](https://www.moddb.com/mods/bannerlord-online/images/image-2#imagebox)
Bannerlord Online [20] is a complete overhaul to the original game with the goal of creating an almost complete Massive Multiplayer Online Role-Playing Game (MMORPG). It uses the main campaign as the basis of the game and synchronizes the game with every player playing in the world map.

As of October 2021, the game has several servers distributed worldwide, a new player can enter the game by choosing the server of choice and registering a new account. The player then creates a character similar to how they would do in the original game and then spawn in a world with other players simultaneously playing.

The map is open world and takes place in the original map of the base game, although it is a bit barren compared to the original. The player will be sharing the world with multiple other players that are roaming the map, they can interact with each other and can even battle NPC armies together; Player versus Player (PVP) battles are still not implemented at the time of writing this thesis but it has been announced that it will be present in future updates.

The player can do a few basic actions like caravanning goods from cities to cities, fighting looters, tending farms to earn money and interacting and socializing with other players in the world. The player can also battle other players in the arena (the current only form of PVP in the game), it works like a royal rumble were all players must kill each other and be the last man standing.

### 3.4.2 Bannerlord Co-Op

![Two clients connecting to a server](https://www.youtube.com/watch?v=Iyb_Au9iJKA&ab_channel=RHCOMMUNITY)

Much alike Bannerlord Online, the mod Bannerlord Co-Op [19] aims to allow synchronous play between players. But rather than being an MMORPG it is a CO-OP, which is usually composed of 2 to 4 players that play the game in cooperation.

---

*6Taken from https://www.youtube.com/watch?v=Iyb_Au9iJKA&ab_channel=RHCOMMUNITY*
In this mod’s case, the goal is to have two individual player parties roaming the same open world. They can help each other mutually in battle and both players can play the normal game together.

The mod is still in early development and is currently trying to acquire more software developers to expand the scope of the project.

At the time of the writing, October 2021, the project is open source [18] and available online, and as of right now, it is possible to set up a basic connection between two players; although not much can be done, the players can see each other in the world and can walk around whilst maintaining connection. No gameplay is possible and the mod is yet to be released in a playable version.

3.4.3 Conclusion

Whilst both these mods are still fairly early in development compared to the final potential, and only one game being currently available for public playing, these mods serve as proof and example on how to do a multiplayer mod for Bannerlord without requiring multiplayer modding tools. However they also show that achieving synchronous battling isn’t feasible for a single student to do in a few months. These mods serve as an example that synchronous play (playing simultaneously in the same session with other players) stands as too out of scope of the thesis and for this reason the General Lord mod will only support asynchronous battling (a multiplayer mode where each player battles in a session by himself).
Conceptual Model

Contents

4.1 Overview ................................................................. 39
4.2 In-game loop ............................................................ 40
4.3 Meta-game loop .......................................................... 43
4.4 Player Progression ....................................................... 49
In this section we will describe the mod's core mechanics and how its features interact with each other.

4.1 Overview

![Figure 4.1: Relation between the in-game loop and the meta-game loop](image)

![Figure 4.2: Main Screen](image)

39
The mod developed over the course of this project is composed in its essence by two forms of gameplay progression which intertwine with player progression. These two systems are the in-game loop and the meta-game loop of the mod. In addition an elo rating system was implemented that serves as a metric for the player progression, as mentioned in 2.1.3.

The in-game loop essentially consists of a combat system that has a character controlled by the player and an army for him to control, the player will fight the battle and either win or lose based on the performance in battle. After each battle the player will enter the meta-game loop. The character that the player controls and his army are evolving features, and this evolution takes place in this loop. It's also here that the player will select his next opponent and enter the in-game loop again. The player can challenge either available equivalent opponents (in ranking and / or army strength) or select from the top 20 players in the world, if he feels like he can match them.

Unlike the original game, there is no concept of borders or territory in this mod, the player goes from the main interface to the battle where he will find himself in a plain with his army close by and with the enemy army further ahead. After the battle is won he goes straight back into the main interface where he can make his decisions. So a constant cycle is made between the in-game loop and the meta-game loop. And alongside these loops will be the constant presence of the player progression. The player will get better at fighting battles and better at managing his army over time, so as the complexity of the game increases so should the overall capability of the player.

### 4.2 In-game loop

The in-game loop of the mod is composed of a combat system, where the player battles an army using the original combat system. Essentially he will bring his army and play as his character and face the opponent's army and character that is being controlled by the AI. In this combat, besides the standard mechanics that the original game already offered, such as commanding his troops and positioning them, the player also has access to combat plans and formations which will be explained further down the document.

#### 4.2.1 Combat and Battles

Combat takes place in a grassy plain where the two armies will face each other in battle. The armies would be randomly positioned in said plain but would always be facing each other.

The player enters the combat controlling his character. It plays like a third-person game similar to how RPGs play like. This character will be wearing the gear and mount that was selected previously during the management phase.
The player’s army will be somewhat near the player in a default formation awaiting the player’s orders. The player can give a full array of commands to his troops, he can order their position, formation (line formation, shield wall, circle, etc.), movement speed (such as advance, charge, etc.) and can even give more complex commands such as telling units to skirmish and maintain distance of the enemy, or the player can simply command the troops to automatically designate their own orders. All these commands are part of the original game combat loop.

In this mod we also added two functionalities: Army Formations and Battle Plans.

### 4.2.2 Army

The army is composed of individual troops, meaning a soldier can perform autonomously and has its own agency. The troops fall into four different categories: infantry (the main body of the army), archers, cavalry and horse archers.

Even though each soldier performs autonomously, they are assigned to a formation (a group of soldiers) and it’s through these formations that the player commands his army. There are eight possible formations, however usually only the first four are used and are composed of each troop category respectively, as previously mentioned.

### 4.2.3 Army Formations

Like mentioned before, when the player enters the battle his army will always be positioned in a default formation. Although the player can manually set up where his troops should be and what form they should have, this can be time costly and usually the enemy party is already charging on towards the player, giving the player little time to prepare his army formation.
The army formations feature was adapted from the thesis “Cunning Lords: Tactical AI and Troop Control for Mount & Blade II: Bannerlord” [12], by Joaquim Quadrado, and was integrated into the mod. The feature allows the player to first define the desired position of his troops and then he can save the desired formation into one of 3 available loadouts. Then when the player begins a battle he can just press a button and all the army will automatically position into their respective positions.

The player can save the loadouts using Ctrl + F10/ F11/ F12 (if he wishes to save to the first, second or third loadout respectively) and load the loadouts using just F10/ F11/ F12 to load these loadouts in battle. This saves the player time and effort and allows him to further customize the overall combat loop.

4.2.4 Battle Plans

Selecting where the troops should be positioned wasn’t all the player had to do. The player also had to either manually command all his troops at the same time (and personally fighting the battle too) or he could give certain or all formations personal autonomy to do as they please. Meaning the player had only two options: either fully micro manage the army or completely relinquish control of the army or part of it.

Like the formations, this system was also adapted from the thesis “Cunning Lords” [12] and is called the Battle Plans. These allow a middle ground, where the army is being autonomous but is following the strategy laid out by the player previously to the battle. It allows the player to prepare a battle plan from start to finish.

The system is divided in five different phases depicted in the appendix A.1. In each phase the player will define how each formation in his army will behave using one of the eight available orders also depicted in the appendix A.1.

This allows the player to decide how his army will approach the enemy, who engages, who flanks, when does his army advance into melee combat, etc. It allows for the full customization of his army behaviour without requiring direct command from the player. But if needed, the player can always assume command of the army at any time.

4.2.5 Defending

Just like the player can challenge other opponents, he himself can be challenged for battle. The opponent will face a representation of the player’s army controlled by the game AI.

To further customize how the player army behaves, much like the battle plan he can use for his own combat, he can setup a defensive battle plan that executes against the opponent when the battle is taking place. The goal was to give some agency to the player on how his defense would occur, by setting up his army behaviour we could give some influence to the player and hopefully attenuate the
fact that he couldn’t personally defend this attack.

The player also has the option to select one of his formations to be fielded during the defense. The army will first deploy into formation and then will execute the battle plan as expected.

### 4.2.6 Results and post battle

After the battle the player should expect rewards based on the performance of the battle. The player will receive gold and experience whether he loses or not based on the events in the battle. But he will receive nothing if he chooses to retreat rather than seeing a battle until the end.

This is due to the fact that retreating as opposed to fighting till the end has too much of an advantage as it avoids more troops from becoming wounded, allowing players to quickly challenge another player. There could be scenarios where certain death is guaranteed at a certain point, but at the beginning he has the upper hand by having good ranged units, he could take potshots and kill some troops, retreat and earn the gold achieved in the battle without too much cost. So if the player really wants to earn the gold, he has to complete the battle giving it an important risk reward factor. If the player wins, he will also earn renown based on how hard the expected battle should be. And finally he will also receive or lose elo which will be further explained in the next section.

### 4.3 Meta-game loop

Outside of combat, the player will then enter a loop where he must decide how to progress and his next steps. This is our Meta-game loop where the player should use his resources earned from battle to further progress his army by either recruiting or upgrading his troops, buying better gear for his character, etc.

#### 4.3.1 Army Progression

The player’s army is the most complex structure of the game. The player can recruit new troops using the gold he has available; like mentioned before, troops are individual entities so the player is always recruiting one soldier at the time. After the unit has fought enough battles, it can be upgraded also using gold.

In *Bannerlord* the troop categories are mixed in together in a single upgrade tree (there are multiple upgrade trees for different cultures). The player usually doesn’t recruit full fledged archers or horsemen, they all start as low level recruits from the same troop tree and is up to the player to decide where a particular soldier will specialize. Each specialization has a tier, recruits are tier 1 and each upgrade goes up a tier up until tier 5. Usually a troop specializes in either melee (infantry and cavalry) or ranged
(archers and horse archers) by tier 2 and by 3 and 4 it goes specifically into one of the previously mentioned.

To further emphasize the need to evolve the army the player can only recruit the tier one troops (often designated as recruits); these troops are poorly equipped and have poor morale as they often run away early from the battle. After a battle if the troops have acquired enough experience the player can spend gold and upgrade them into further tiers up to tier 5. After a while the player can unlock noble troops trees, that are better than the regular ones and start immediately in the tier 2 and are the only ones that can upgrade into tier 6.

Some specific troops require extra items (and therefore extra money) to upgrade to a certain tier, these are mainly the horsemen as you have to buy them a horse or a better one from the current one. Horses are equipment that can be bought with gold.

4.3.2 Wounded Troops and the Infirmary

In this mod the concept of casualties was fairly changed and adapted to the constraints of the mod. In the original game when troops fell in battle they could either die (permanently lost troops) or become wounded (temporarily lost troops). The latter would then recover over time (in game time, meaning the player needed to play the game unpaused to recover these troops) and be added back to the “normal” troops that were combat ready.

In this mod we opted to remove the concept of dead troops and revamp the concept of wounded troops. We felt that permanently losing troops could be too harsh to the players, as the only way to earn money in the mod is to fight and kill enemy troops, therefore the player could eventually find himself
locked out of the game without money to recruit new troops and fighting battles alone could be too hard to earn any money.

To achieve a balance were the player could play the game for a fairly amount of time without having to be forced to pause too early we decided to have a fraction of the fallen troops in battle to be immediately recovered and the other fraction to be wounded and therefore temporarily removed from the party. However, since no in-game time is passed, the troops are locked by real time. After each battle, every troop that was designated to be wounded will be removed from the party and added to the “Infirmary” for half an hour (each individual group of troops lost after a battle has a cooldown of half an hour until after said battle). After the troops are recovered they are added to the garrison, where the player can retrieve them back into the army should he wish to do so. The Infirmary itself serves only as a screen that lets the player check who is currently wounded and has no direct gameplay impact.

4.3.3 Garrison

The player doesn’t have to bring every troop to a battle, he can better specialize his army by leaving some troops behind if he so desires. This allows the player to, for example, create a fully cavalry army for one battle and in the next one have infantry and archers joined in the mix. The goal is to give the player the freedom and choice of his army composition.

Allowing the player to customize his army and army size also helps him to challenge weaker armies or try to gain more elo (which will be explained further down the document) than he would get if he had his full force in the field.

4.3.4 Party Capacity

The player can only bring a specific amount of troops to a battle. He starts with a small limit of only 20 troops (his character includes as one of those, therefore he can recruit 19 more) that can be increased by either improving stewardship or moving up in clan tier. Both of which will be explained in further sections.

This forces the player to be tactical in which troops he brings to a battle. The player can add infinite troops to the garrison but he can never add more that the limit to his party, so even though he might have hundreds of troops in garrison he won’t be able to bring them all unless he upgrades his party capacity using the aforementioned mechanics.

4.3.5 Clan Tier

Clan tier is a soft measure of the player stage in the game. The clan tier ranges from 0 to 6 and with each tier the player receives more party capacity to field a bigger army on the battlefield.
At specific tiers the player will unlock more troops and equipment for the player’s character. At tier 0 the player only has available recruits from the culture the player selected for his main character. At tier 1 he unlocks every other culture’s recruits and equipment. At tier 2 and 3 the player unlocks rarer and stronger troop types, such as the vigla recruit that eventually could upgrade into the imperial cataphract, the strongest cavalry in the Empire.

The player can upgrade his clan tier by receiving renown. Renown is a mechanic that represents the value that tracks the progression of the clan tier. The main way to earn renown is by fighting hard battles when selecting the opponents: the harder the battle, the bigger the renown reward. This is still fairly slow and if the player wishes he can invest gold to buy renown to speed up the process.

4.3.6 Economy

The economy is a crucial part of the meta-game loop, it helps the game flow and progress at a steady pace and to soft lock mechanics from the player until he is ready to interact with them. Gold is the virtual currency of the game and is earned through battling, it is mostly used to upgrade and improve the player’s army.

The player earns gold from challenging other players and killing their troops. Each troop can drop between 40 to 45 gold times the troop tier (from 1 to 6), which means stronger, harder to kill troops yield more gold. The scaling of gold yield helps the game scale and to soften the prices of items and troop upgrade over time. This also incentivizes the player to try and battle harder enemies because the reward will be greater.

There are a lot of ways to spend the gold in General Lord. The main one is recruiting and upgrading troops, as this is the most vital mechanic in the game. The cost of recruitment and upgrade were increased in the mod mainly due to balance issues compared with the original game. In the original game, the costs to recruit and upgrade are cheap for two main reasons: Troops die, making the investment lost easily, and there is a daily (in-game time) cost called party wager that scales with troop count and troop quality. In the mod none of these are present, once acquired these troops are neither permanently lost, neither have any constant cost associated with them. For this reason the price of recruitment and upgrade was increased 10 times the original cost. This forces the player to better decide where to spend his money because upgrading troops actually means an investment rather than an easy spend given how cheap they were.

In the shop section the player will have access to a plethora of weapons and armor that he can buy to equip on his main character. The items are quite expensive and increase heavily in price the better they are, so it is likely the player will only start to upgrade his main character after he’s passed the early stages of the game; this helps the player focus more on improving his army as it is the core mechanic of the game mode. These items can be sold back for a fraction of the price, so if a player buys a new
weapon and he doesn’t like it or eventually he gets a better one he can always get some value back from his investment.

The player can also spend his money training his stewardship for 50 gold each time. Stewardship doesn’t have a colossal impact in the game as it pretty much only improves the party capacity, so being cheap for a low value return seems appropriate. The price inevitably scales as each button press yields the same experience, but the experience needed to level up increases with each level, requiring then more clicks to level up each time.

Similar to stewardship but considerably more expensive, as it yields more rewards, is the button to buy renown at 250 gold per click and earn 10 renown. Clan tier renown threshold scales considerably each tier, so the cost return of this button decreases over time, requiring more gold to be spent per tier.

If the hero isn’t full health, the player can choose to go to a doctor for a fee to fully heal. This fee scales based on how much Health Points (HP) the player has to restore, each HP costs 5 gold.

4.3.7 The Player Character

In the original game, the player also has a character that progresses and gets stronger over time. The character has skills and proficiencies that can be developed by performing the required actions for it; for example, hitting enemies with a bow will improve the archery skill and eventually unlock new perks that makes the player or army stronger.

The player character also has equipment that can be upgraded for better gear. This ranges from helmets to boots and from swords to most kinds of medieval weapons. The player also has access to horses and horse armor.

In the mod, most of these features were incorporated into it and adapted to function properly in the mod’s environment.

4.3.7.A Skills and Perks

All skills of the game are present in the mod, although some of them have no use and aren’t relevant to it. The player will focus on the combat mechanics, as all the social skills are not important to the game mode; these skills couldn’t be upgraded either way as it isn’t possible to perform the actions required to upgrade them.

The only skill we decided to adapt was stewardship, because this skill is required and helps the player increase his party size. To attenuate the lack of actions to increase the skill’s level, a button was added that allows the player to train his skill in exchange for 50 gold.

After every few levels in each skill, the player is presented with one or more perks to unlock (in cases where multiple perks are available, the player must choose one of them). These perks are specific bonuses to the player’s character and some even apply to the whole party. Just like the skills, some
perks are not available to the player as the player cannot unlock them, since he can't upgrade the skill that unlocks such a perk.

### 4.3.7.B Equipment

All equipment works exactly as the original game does. However, the way to acquire said equipment had to be adapted. The player will buy his equipment in a designated shop with most equipment available for him to choose. Some items are locked initially due to Clan tier. Unlike the original game, no equipment is earned as loot after winning battles.
4.3.7.C Character Healing

After each battle the player is prone to take damage on his character. In the base game, when the character loses HP in battle it remains the same after it concludes and has to wait in game time to recover them. The player can upgrade his medicine skill so he recovers faster and unlock certain perks that allow him to recover some health immediately. If the player falls in battle he will be unable to command his troops for a while until he recovers enough health.

In the mod rather than tying player recovery to in game time, the player recovers always 30 percent of the max health after each battle, or if he is close to full health he gains enough to reach the max. Any leftover points can be bought by going to the doctor for gold.

By always leaving the player with at least 30 percent health, the player is always able to command and lead his army in battle, but won’t necessarily be in the best conditions and could be easily taken down if not careful. When you fall in battle you become unable to control your units, until the rest of the battle, so the player has to think twice before charging the enemy with low health.

By locking the rest of the recovery by gold, it serves as a money sink whilst presenting a choice to the player of either saving gold and risking fighting on lower health or by spending it and be on the safe side.

4.4 Player Progression

Since the mod heavily relies on the original game’s combat mechanics, the player’s combat skills and strategies are naturally tied to his previous experience on the game.
New players should struggle a little at the beginning until the basics of combat and managing the army on the field of battle are understood.

The mod allows free reign on how the player wishes to progress, be it in the in-game loop where the player defines his strategies and approaches to combat, be it in the meta-game loop where the player defines what army to use or how to upgrade it. This means that a lot of strategies and compositions are possible and the player should decide which play style fits him best. The player could for example fight solo against hordes of enemies if he has the skill to match, or create a full horse archer army that fights by peppering down the enemy with arrows from a distance.

4.4.1 Ranking System - Elo

The mod and the game can be quite challenging and have various degrees of mastery. For that reason a ranking system was implemented to differentiate the various players from each other, as well as to create a degree of competition within the game mode. This system is represented by Elo [6], an assigned value that tries to represent the player's average capabilities compared to other players. When a player wins or loses a battle his elo is updated accordingly to the result of the battle, and the amount changed is depending on the expected difficulty of the battle (based on the opponent's elo and army ratio compared to the player's).

4.4.2 Elo formula

The elo formula was inspired in the score formula implemented in João Morais’ thesis [10]. The formula used in the “My Army” [10] thesis is the Elo Rating System [6], which is also used in professional chess.

The formula consists of the following:

$$\text{newElo} = \text{PreviousElo} + K(r - e_{\text{player,vs,opponent}})$$ (4.1)

Where $r$ is the outcome of the battle, a value of 1 for a victory or 0 for a loss. $e_{\text{player,vs,opponent}}$ is the expected outcome based on the past results and the power balance between the armies and is a value between 0 and 1, and $K$ represents the overall maximum elo that can be earned or lost depending on the result of the battle. The elo weight, $K$, was split between an offensive battle or a defensive one, where the player can earn or lose up to 50 elo in an offensive battle (equal battles are worth 25 elo) and earn or lose up to 10 in a defensive one (equal battles are worth 5 elo). This difference was made so that players don’t feel unfairly punished for not being able to defend external attacks as they can’t personally defend them.

The expected outcome $e_{\text{player,vs,opponent}}$ is calculated using the following formula, which is based on
the logistic function [5]:

\[ e^{\text{player vs opponent}} = \frac{q_{\text{player vs opponent}}}{q_{\text{player vs opponent}} + q_{\text{opponent vs player}}} \]  

(4.2)

The \( q_{\text{player vs opponent}} \) and \( q_{\text{opponent vs player}} \) are the respective quality of each player compared to their opponent. The formula to calculate these qualities is the following:

\[ q_{\text{player vs opponent}} = \frac{10^{\text{Elo}/L}}{\sqrt{\text{armypower}_{\text{opponent}}/\sqrt{\text{armypower}_{\text{player}}}}} \]  

(4.3)

Were the numerator comes from the logistic function as depicted in [6], the \( L \) constant serves as a scaling factor to reduce the weight of the elo in the formula. In addition we also need to introduce the army power (a value calculated by the game that takes into account troop quality and troop quantity that determines the overall strength of the army) of the player compared to the opponent as it will affect the overall expected “quality” of the player, a player with a stronger army than the opponent’s army should in theory have better chances of winning regardless of skill. Therefore the value of quality is divided by the ratio of both armies, a stronger opponent will reduce the overall expected quality of the player and a weaker opponent will increase the overall expected quality of the player. We will discuss balance changes to the elo formula in chapter 5.

### 4.4.3 Matchmaking and Ranking Display

![Figure 4.8: Example of opponents list](image)

The player can challenge players via two selection screens. The Match Making screen and the Ranking screen, the latter also has the goal of showing the top 20 players in the game (sorted by elo).
An opponent is eligible for challenge if the player hasn’t won a battle against him in the last 12 hours. This means a player needs to wait 12 hours after each win before trying again, this way the player can’t abuse easy battles that he found he can win and is compelled to try and challenge new armies. If the player loses however, he can try again until he either gives up or eventually wins against said opponent.

The match making screen shows available players based on their average strength and elo compared to the player. The match making tries to give an assortment of equal, a little stronger and a little weaker players. This gives the player room to be creative on how he wants to progress, maybe he wants to challenge weaker players to avoid losing as much troops, although the rewards will be lesser than if he challenges harder, stronger players. This risk-reward factor keeps the player engaged and analysing and scouting the armies previous to battle gives another layer of strategy to the game.

4.4.4 NPC Armies

Besides the armies created by players there are NPC armies that are static and have varied degrees of strength. Some armies serve as the stepping stones for newer players. This way even if a player arrives late to the game and every army is already too strong he can always have 3-4 fights winnable within the first minutes of the game.

These armies follow a theme: the weak armies are composed of looters only. The medium armies are sea raiders (which are barbarians better equipped and stronger than the average looter). Lastly there are three harder ones: the khuzait raiding party, a forty strong cavalry party which only players in a farther stage can reasonably win; the Archon Milus, an imperial general commanding a semi elite force of legionaries, elite archers and cataphracts; and the khuzait horde, the hardest challenge, a force to be reckoned with, a 200 cavalry strong army composed of heavy and light cavalry and hundreds of horse archers. These armies give more flavor to the game and at the same time maintain a simple progression ladder for players, were they can have a fun battle with at least one army in every stage of their army strength.
Development and Implementation

Contents

5.1 The Overall Structure .............................................. 55
5.2 Implementation of the Structure ................................ 56
5.3 Interface ............................................................... 56
5.4 Multiplayer and Constraints ....................................... 59
5.5 Most Common Multiplayer Interactions ....................... 62
5.6 Formations and Battle Plan ....................................... 63
5.7 Game Balance Changes ........................................... 64
5.8 System Limitations ................................................... 66
The mod’s development and construction is set between February and August of 2021. The first three months were spent getting a grasp of creating a mod in Bannerlord and investigating the game’s architecture to plan where to begin when creating the mod.

Given the lack of clear material and tutorials for modding at the time (mainly only one series was present at the time), most modding was learned by reverse engineering other similar mods to understand what their creators did.

By May full development of the mod was ensued which lasted until about July. The month of August was spent bug fixing and implementing the last key features and the game was launched to the public August 26th.

In this section we will give a brief overview of the development itself and the architecture of the software implementation. We will proceed by analysing the interfaces created and how the application was structured around the game mechanics.

5.1 The Overall Structure

The essential structure of the mod, as can be observed in the diagram of figure 5.1, is composed in the following way:

- **The main interface**: A central hub of operations where the player makes most of the management of his army;

- **The battles**: The core loop of the game, where the player battles against the enemy armies;

Figure 5.1: Overall Mod Structure Diagram; Legend: Yellow - Local Systems, Green - Data Shared, Blue - Remote Systems
• **The progression**: Every part of the management made by the player out of battle;

• **The player information**: The crucial information about the player and his armies, equipment and others, required for other players to challenge the player;

• **The server connection**: The system in charge of communicating with every player system and sharing the required information with each player;

• **The server database**: The server storage service that keeps every player’s information and battle history.

### 5.2 Implementation of the Structure

The mod relied heavily on the integration of the combat loop with the progression system present in the single-player, so the easiest way to merge the combat loop with the progression was by using the single-player campaign itself in our favor. The mod is essentially built into the campaign system which allows for easy access to the progression systems of the campaign without implementing them from scratch.

For the battle we used the battle system of the campaign system. Essentially we instantiate a party in the map using the troop information of the opponent received from the server. We then start an encounter in the map between the player’s party and the party representing the opponent.

To achieve the goal of having a stand alone game mode, we hid the single player campaign with the main interface. Essentially showing to the player only the intended features for the mod, allowing us to control the flow of information and removing the unnecessary and undesired features.

Since this battle relies only on the competence of the AI to fight for the challenged opponent and not the opponent itself, we decided that to make the game more fair by maxing out the difficulty of the game in order to make the battle as fair as possible, even so the AI on the hardest difficulty still hardly matches a competent player.

### 5.3 Interface

Like mentioned in the previous section we used the main interface to hide the single-player and easily control the flow of information to the player. For this it was required that the main interface was complete but still easy enough to read. The goal of the screen was to serve as a base of operations for the player and a link between the battles and the progression system, making each system of easy access.

The screen is divided into three main parts, the top side, the left side and the right side as depicted in figure 5.2.
Figure 5.2: Main interface divided in three sections

The top side contains easy access to most forms of the meta-progression loop of the game; it also displays some information such as the clan tier on the right side and the character information on the left. In the middle it also displays the mod title where if you hover over, a brief explanation of some game mechanics will pop over. The five buttons available are the following:

- **The Character Screen:** The character screen is exactly the same as the one present in the original game, it allows the player to see his character stats and skills as well as allocating the focus points (that increase the skill progression speed) and choosing perks.

- **The Party Screen:** The party screen also works exactly the same as the original game. It allows the player to check his troops, upgrade troops to the next tier if they have the XP necessary and if the player has the gold, and allows the player to dismiss troops he doesn’t want in his army anymore.

- **The Garrison Screen:** The garrison screen was adapted from the party screen and allows the player to store troops into a different roster effectively storing those units. The player can also retrieve any unit that is stored as long as he has the party capacity to do so.

- **The Recruitment Screen:** Much like the garrison screen, the recruitment screen was an adaptation of the original party screen. It works a bit differently though, as in this screen the player isn’t able to shift any troops from the right roster (his party) and is only able to add troops from the left
roster (the troops for sale) if he has the money to add them. The decision is only executed when the player presses the done button, until then any purchase or action can be reversed using either the reset or cancel button.

- **The Shop:** In the shop the player can buy equipment for his character; here he can manage and change the equipment in his inventory, he can also sell bought equipment if he so wishes. It’s also here that the player can buy horses to upgrade the cavalry units.

In the **left side** of the screen, the player will see a preview of his character and beside it he will see most important information for most player’s decisions. This information contains the following:

- **The Player Party Information:** Information on the state and composition of the player army. First is the party name and right below is the party size and capacity. Next is the total army composition, which contains all the player’s troops, whether they being in the party, garrison or wounded.

- **The Player Elo and Gold:** Then he has the current elo and gold in display. The gold will most likely be a source of decision making for the player, if he has low amount of gold the player is likely to go to battle to earn more and if he has a high amount he is likely to spend the gold to upgrade his party.

- **The Skills:** Next to the player character preview there is also a display of the current skills of the player.

- **Essential Information For Battle:** Below is the information of the battle ready troops, how many and what are these troops (essentially a quick way to check what troops are in the party without opening the party screen) and the player’s current character health. The goal is to give the player quick information on the state of his army, so he is able to perceive the situation properly and make decisions based on it. Should he go to battle again? Or is it time to upgrade his army? Or is his army too depleted for another battle and he has to rest?

- **Leave and Save Progress:** Lastly at the bottom is the save and exit button that allows the player to quit into the main menu.

The **right side** of the screen is more focused on the management side of the game. It contains more information regarding the meta-game loop on the top side and the buttons to begin battles on the bottom. The overall structure is the following:

- **Garrisoned and Wounded Troops:** At the top is displayed the amount of troops garrisoned and wounded as well as the composition of the troops. A bit lower there is also a button to personally check the troops wounded, it works like the garrison screen but the player is not allowed to exchange any kind of troops. It also has a counter showing the next batch of troops to be recovered as well as how much time is left for them to do so.
• **Transaction Buttons:** Below are the buttons to heal the hero, buy renown and train stewardship along with cost to perform each action.

• **Battle Plan Buttons:** Next are the buttons that allows the player to customize the battle plans (opening them shows a new screen detailing each order for each formation in each stage). And below it there’s a toggle button to decide if the AI should use the defensive battle plan when the player is attacked or not.

• **Ranking and Find Opponents:** Lastly there are four buttons at the bottom of the screen: Ranking, Match History, Find Opponent and Formation Manager. The buttons ranking and find opponent opens a list with opponent players’ armies detailing their elo, army composition and overall strength. The player then has the option available to challenge each army. In the case of the ranking screen, only the top players are shown and they are shown in order. There are also four buttons at the top that allow the player to sort by either: name, elo, army power or troop count. All four have the ascending and descending order available (order is changed by clicking again on the same button).

• **Match History:** In this screen, all the matches of the player are displayed with information of the offensive and defensive battles of the player. A match history entry has the following information: Date of battle, the result of a battle, who were the participants and their initial elo (elo before the battle), and what was the elo change for the player. Each entry is also clickable and drops down additional information such as the overall army strength of each army, the army composition of each player and the casualties of each player in the battle.

• **Formation Manager:** This button launches a mock battle were the player can define his formations and save them.

### 5.4 Multiplayer and Constraints

Last semester during the development of the thesis’ project, multiplayer in *Bannerlord* was already present but it wasn’t yet fully developed and multiplayer modding was still in question if it would be supported or not. Several months have passed and there wasn’t much progress since then.

As of October 2021, multiplayer modding is still not supported. So we had to improvise and looked onto other similar mods such as *Bannerlord Online* and *Bannerlord Co-Op* and how they are doing their multiplayer connection. Essentially both created server instances that communicate with their mod.

Since no support or help would come from the game into implementing the multiplayer, synchronous battles quickly were discarded as they would be too complex to implement by one student only in just a few months. So it was decided that only asynchronous battles would be implemented in the game.
It was at this point that the overall multiplayer experience would be defined: an asynchronous experience where the player would challenge opponents available online. For this the system design would be a simple server that would serve as support to the main game and provide the game enough information to present this experience to the player.

Thus, we needed a server capable of storing the players’ armies and provide their composition to other players when they decided to challenge them. The game would then recreate the opposing army and use the system described in section 5.2 to instantiate a battle using the opponent’s army. This effectively allowed the player to fight the opponent through the means of an AI controlling their army.

So for the General Lord’s mod, a small prototype of a server was created. Since the main focus of the thesis is to observe persistent progression in multiplayer and not the multiplayer experience in itself, there wasn’t much focus in securing the server with player authentication or making most of the logic server side rather than game side. The focus was ensuring the player could battle many armies and to have a complete prototype of the mod’s multiplayer experience.

### 5.4.1 General Overview

![Server Client Connection](image)

**Figure 5.3:** Server Client Connection; Legend: Yellow - Local Systems, Green - Data Shared, Blue - Remote Systems

The server is a simple ASP.net (“ASP.NET is an open source web framework, created by Microsoft, for building modern web apps and services with .NET.” [9]) server with connection to an SQL (“domain-specific language used in programming and designed for managing data” [2]) database.

It connects to the game via Application Program Interface (API) calls. Essentially every stage of the game that requires external information (for example: finding the available opponents or calculating the elo gained after a battle) it will call upon the desired method using the specific url and the server will execute the method and if needed it will respond back with the gathered information. The server has two main purposes: Store information and distribute said information. And as such, most calls do one
of those specifically.

5.4.2 Information Stored

The server stores information needed to be shared with other players and to validate some information of the player itself. The profile is the representation of the player’s information and contains most of the required information to properly make a connection between players. The information is the following:

- **Id and Unique Id:** When a profile is first created, the player receives a new id and if this is the first time playing the game he will also receive an unique id. The id is the identification of the character and most operations are done using the id, each id is associated to one and only character. The unique id is unique for each player, meaning it is associated to every character created in the same computer, this allows the player to play multiple campaigns and at the same time avoiding players to create multiple accounts to battle and evolve against himself.

- **Player Information:** The profile stores the name of the player’s character, the army strength and total count of the player’s party, the current elo of the player and the entire army composition of the player’s party.

- **Formations and Battle Plans:** It also stores the defensive formations and battle plans as well as if the player has them enabled or not.

- **Game Metrics:** Several Game Metrics are stored such as how many times a player presses certain buttons, how much gold he has earned or spent, how many times he opens or closes the game and how much time he has played.

- **Wounded and Garrisoned Troops:** It is also saved in the sql database the current wounded and garrisoned troops.

The **match history** entry represents the overall registry of each battle of the player. The match history, besides being a record for the player to see the battles won and lost, also has a lock mechanic for the match making: if the player has any victory against the opponent in the last 12 hours, then he cannot challenge said opponent. The information stored in the match history is composed of the following:

- **Player Id and Enemy Id:** The id’s of both players serve as a quick way to identify to whom belongs this match history.

- **Player and Enemy Information:** The match history entry also holds a lot of information related to the participants: both player’s character name, the army strength, the troop count and their
current elo. It is also saved for reference what troops went into battle from each side and how many casualties happened for each side.

- **Battle results**: The match history also carries over the information about the result of the battle (if it resulted in the attacker victory or loss), the date and time of the battle, and the change in elo for both sides.

### 5.4.3 Other Practical Uses

The server gathers and stores critical information for the correct functioning of every player participating in the mod. This also allows for statistical data analysis of player information such as the time the player spent playing, how much gold he earned or spent, the composition and size of his standing army, wounded army and garrisoned army, etc. So besides being a backbone to the mod’s functioning it also gives some data for analysis.

However, there was an oversight in the implementation of this mod as a permanent log of each player’s profile entry update was not created, and instead we only had access to the current most updated version of each player entry as all others were overwritten. To attenuate this issue, a daily screenshot of the server’s database was done to keep track of player’s progress and behaviour.

### 5.5 Most Common Multiplayer Interactions

Over the course of the gameplay the player is bound to make actions that lead to a lot of calls to the server. The most common is a call made to update the player’s profile.

#### 5.5.1 Player Profile Update

A balance had to be achieved when it came to server calls to update the player profile in the server. Not every change in the game needs to be stored or at least immediately stored.

To ensure the profile is correctly up to date when needed, profile updates are called every time the player saves and exits, and when any decision and gold spent by the player has been made permanent. The profile is also updated every time the player finishes a battle, because naturally his army will suffer changes such as wounded troops being added or the money he earned from the battle.

Other interactions that are not necessarily crucial decisions don’t need to be immediately added to the server. For example, if a player chooses a new perk or improves one skill, it will be updated either when a crucial action is actually executed or when the player saves and exits the game. The reason is that even if the player leaves the game without saving (for example ALT+F4) the change was not saved
in the game and therefore doesn’t need to be saved in the server, the player could in theory open the game, do the same action and eventually save when he wants.

5.5.2 Match Making Selection

To show the player the eligible players for challenging, a query is made to the server where an algorithm is executed to select the most appropriate opponents for the player. This allows to maintain interesting battles were the opponent is neither too strong or too weak.

The first criteria was whether the possible opponent had already been recently challenged by the player in question. For this we searched the player match history and if he had any recent (less than twelve hours) offensive victory against that opponent he would be unable to challenge said opponent.

The second criteria was if the unique ids of the opponent and the player were different. This was made to avoid players creating new characters to be used to improve against themselves.

The third criteria was if the opponent was within a certain threshold of relative strength (a balance between elo and army power) compared to the player. The formula was the following:

\[
\minThreshold < \frac{(Elo_{player}/L + armypower_{player})}{(Elo_{opponent}/L + armypower_{opponent})} < \maxThreshold
\]

(5.1)

Where \( L \) was once again a constant that serves as a scaling factor to reduce the weight of the elo in the formula to be similar to the armypower. As an initial test we decided to give equal influence between the elo and army power in the ratio, during testing period the formula appeared to remain appropriate. The minimum threshold was set at 0.6 and the maximum threshold was set at 1.6, meaning the player must be at least 60% as strong as the opponent and can’t be stronger that 160%. Once again, these values were what seemed balanced enough to give a healthy amount of opponents of similar strength to the player to challenge.

5.5.3 Post-Battle

After a battle ends, information is sent to the server on the results of the battle, how much troops were in the battle, how many of them were killed or wounded and other necessary information. The server will then create a match history entry in the server database with the data of the battle as mentioned in section 5.4.2. It will also update both participants profile with the respective elo change from the battle.

5.6 Formations and Battle Plan

Like mentioned previously most of these features were implemented by Joaquim Quadrado in the thesis “Cunning Lords” [12]. He then provided me with most of the features and these were later adapted by
me to also work on the adversary, thus creating the possibility to have defensive formations and battle plans (meaning plans that work when you are attacked).

The defensive plans were implemented mostly by storing the information on the server (if a plan / formation should be used, and if so, what is said plan / formation) and then when a player is attacked, the other player's game will receive this information and implement those battle plans on the adversary team if needed. This way the players can fight against plans detailed by other players and can use their own if they wish to do so.

5.7 Game Balance Changes

There were some balance issues that had to be addressed. Some of them came up previous to the mod's release, whilst others came after the release into public testing.

5.7.1 Upgrade Cost Being Too Cheap

The upgrade cost in the base game was really cheap, the only effective major cost for upgrading troops was the horses the player had to buy for upgrading into cavalry units. Like mentioned before this was mainly due to the fact that these troops are perishable and add a constant price to pay each in-game week. To attenuate this factor the price was increased in 10 times the original value. The price is still fairly accessible (for example, recruiting a recruit is 100 gold and upgrading a recruit is 150), but forces the player to actually make a decision if it's worth it to spend that much money now rather than being a no brainer action.

5.7.2 Differentiating Noble Troops from Regular ones

Noble troops are fairly stronger compared to the regular ones, so when presented the choice of either one, the player would always steer toward the stronger noble troops. So eventually we “nerfed” the price of noble troops and increased it by 3 times the original value. Meaning they now cost 300 gold per tier level rather than the original 100.

5.7.3 Elo balancing

The original formula for the $q_{player, opponent}$ and $q_{opponent, player}$ was the following:

$$q_{player, opponent} = \frac{10^{Elo/L}}{armypower_{opponent}/armypower_{player}}$$

(5.2)
Like mentioned in section 4.4.2, the denominator represents the overall ratio between the player’s army and the opponent’s. This formula was based on the implementation of João Morais in his thesis [10].

In later stages of testing we reached the conclusion that the army was impacting too heavily the elo change. A skilled enough player could best armies of 40 or 50 enemies by himself even if he had a bigger elo. This made that the power difference being so larger would yield the maximum amount of elo to the player even though he already had more elo that the challenged opponent. So we had to refactor the formula to reduce the impact of the army on the elo, but still making it a relevant factor when deciding said elo.

We decided to reduce the influence of different armies by applying the squared root to both army ratings:

$$q_{\text{player vs opponent}} = \frac{10^{\text{Elo}/L}}{\sqrt{\text{armypower}_{\text{opponent}}}/\sqrt{\text{armypower}_{\text{player}}}}$$ (5.3)

This ensures the ratio remains closer to one, thus lessening the influence of the value and increasing the influence of the player’s elo history.

### 5.7.4 Unaddressed Balance Issues

There were some other balance issues that were chosen to be unaddressed as we were too far into the testing phase and changing them then could’ve affected the results of the testing phase.

The cost of upgrading clan tier through the “buy renown” mechanic was rather cheap and should probably be increased. However increasing it would seriously affect progression speed of players and would create different results compared to the days prior to the change.

There was also an observed steep difficulty to the early stages of the game. At the beginning of a campaign the player has little gold to spend and will most likely always invest it in upgrading and recruiting troops. But there is a serious limit to how much troops he can recruit as battles yield a low amount of gold at the beginning. In contrast a bad battle could lead to a serious loss of troops, although a temporary loss, would mean the player wouldn’t be able to properly battle until these troops recovered. This could lead to situations were the player would be at a net loss and would take time to improve, seriously slowing down progression at early levels. Another factor could also be the character healing being too expensive for the early stages, either further draining the player’s low funds or forcing the player to go already weakened to battle.

These factors can be frustrating to new players but by the time these interactions were observed, it was decided not to address them, as once again it would seriously affect the results and the progression speed of the players.
5.8 System Limitations

There were some limitations that came from the difficulty associated with modding in *Bannerlord*.

One is the perks system and some skills weren’t fully integrated in the mod. Not all skills were made available, as some would need refactoring or some form of progression that was not included at the time of the mod development. One such skill was the Leadership skill, a similar skill to Stewardship, it had some bonus to party size but much of its perks would be hard to support and we lacked the time to properly implement it in the mod.

There was also a limitation with including the player’s perks when he was attacked by another player. The correct transfer of perks to the server and to another game was not achieved at the time as the game doesn’t allow dynamic hero generation which was required to dynamically assign the perks of the incoming opponents.

Another limitation was the wounded recovery system. Due to limitations of modding and to avoid possible crashes, the recovery was made using a timer and recovering each group of troops as the timer ticked. This system also required the player to be online which means that the player would need to log on, wait a few minutes until all troops recovered and only then would he have fully recovered his army. This limitation could be frustrating to more impatient players and should be refactored if the opportunity arises.

Another limitation that arises from time constraints would be a recording system of defensive battles and allowing players to review their defensive battles. It would be an important feature to complement the defensive plans as it would allow the players to understand how their defensive plan performed and what they could do to improve it.
6

Tests and Feedback

Contents

6.1 Mod Release ................................................................. 69
6.2 Dissemination and Community ........................................ 70
6.3 Mod Reception ............................................................. 71
6.4 Automatic Data Collection .............................................. 72
6.5 Open Survey ............................................................... 80
6.6 Controlled User Tests ................................................... 82
6.7 Conclusion ................................................................. 85
After the mod was implemented and tested for the major bugs, it was time to release the mod into public testing. In this chapter we will discuss the mod release, advertising and community interaction, the results and player activity, and the results of the survey. We wanted to evaluate the overall usability and enjoyment of the mod so we, in addition to releasing it to the public and asking for feedback, performed a series of controlled tests with newer participants and evaluate their adaptation to the game mode.

### 6.1 Mod Release

![General Lords' logo](image)

**Figure 6.1:** General Lords' logo

The mod *General Lords* was launched in August 26th for public testing in the site *Nexus Mods*\(^1\), a popular modding site that hosts mods from several games. The choice of site was made for two reasons: *Bannerlord* being in early access still does not have access to the *Steam Workshop* (the *Steam*‘s modding platform) and *Nexus Mods* is currently one of the most active modding sites available.

#### 6.1.1 General Lords' Page

The *Nexus Mods*’ page, depicted in figure 6.2, serves as a preview of the mod for the general public. The front image that the players will see as well as the name of the mod is the logo created depicted in the figure 6.1, the aggregation of the mod’s name to the game’s original banner.

The page also contains a brief description of the mod as well as some preview images to captivate potential players; it also contains the downloadable files that compose the mod, allowing the players to

\(^1\)Nexus Mods, [https://www.nexusmods.com/](https://www.nexusmods.com/)
then download the files and place them in the Modules folder in their game, permitting them to play the mod easily and immediately.

The page also provides some metrics and stats regarding total downloads and total endorsements (a mark of approval by the players) the mod has received.

The modder can also add some tags to the mod to help it be found by players search specific kinds of mods. The tags added were: **Gameplay**, **English**, **Multiplayer** and **Persistent World**.

The mod page also provides easy version control for the mod and the player downloading will always be aware of what version he is downloading (and if it is the latest).

The mod’s description used in the mod page aims to give a broad overview of the mod whilst keeping it engaging and giving people the desire to try it. It also contains a few warnings so players are expecting and understand possible limitations of the mod. The description can be viewed in the appendix A.2.

### 6.2 Dissemination and Community

To better propagate the mod’s visibility, posts were made in the Bannerlord’s Reddit pages, Figures B.1 and B.2, and the Bannerlord Community Forum, figure B.3, giving a brief description of the mod and incentivising the players to try and to join our community in a Discord server created by me and 4 other colleagues that were also developing a mod for Bannerlord.

Overall these posts provided little results with only 7 upvotes in one Reddit post and 9 in another. As well as a reply depicted in figure B.4. The download rate also didn’t suffer much of a change from the normal rate after creating these posts.

**Nexus Mods** has a feature that allows people to post comments on the mod page as well as submit
bugs for the modder to address and eventually fix. This allowed some users to communicate with the mod creator and give feedback on the mod. However, to better our engagement with the players, we created a Discord server for the Bannerlord Thesis’ mods. This server was disseminated globally over the 5 mod pages created in the Nexus Mods, which allowed some overlap where players first came for a certain mod but also ended up playing others.

In General Lords’ case there was a fair bit of activity and some of the more excited players even messaged privately the mod creator to personally give their feedback and suggestions for the future of the mod. The most common interactions were addressing bugs and questions, appreciation posts and suggestions for further improvement made by the userbase.

6.3 Mod Reception

Despite a small overall userbase we can say the mod had success in some key areas. Over the course of a month it had a steady growth of new players and a few players remained playing the game for quite a while. The mod was also featured in the trending section, figure B.14, and the most downloaded section, figure B.13, of the Nexus Mods’ page.

After the testing period of 30 days, the overall download ratio was at around 325 unique downloads and 510 total downloads (unique downloads representing the number of downloads excluding repeated ones from the same player, which essentially represents the total number of players that downloaded the mod). It had over 15000 views and 16 endorsements (upvote feature of the Nexus Mods).

This means that the conversion ratio from viewership of the mod page was around 2% which was fairly low; this could mean that the overall interest in a multiplayer experience in the modding community of Nexus Mods could be relatively low. However, we also have to keep in mind that General Lords reached the most downloaded page and trending page during the first weeks after launch, which can also indicate that the users of Nexus Mods are probably rather selective in what mods they decide to try. And as of October 27th, 2021, the unique downloads stabilized at 382 and the total downloads at 563. The total number of views reached 18526 and 17 total endorsements.

During this testing period, in the Discord server, we had several of the more active players join to give feedback and appreciation for the mod and the idea. These players also seemed really interested in the mod’s overall improvement and where the most prone into answering the survey. There were also two keen users that made several suggestions in regards of the future of the mod and how to further improve it. These suggestions can be seen in the appendix B: figures B.5 to B.12.
6.4 Automatic Data Collection

During the open testing period, like mentioned in section 5.4.2, the server allowed us to store plenty of information regarding the player's behaviours, although with some limitations as previously mentioned. This allows us to analyse the overall population of the mod and analyse the overall engagement based on the time spent playing the game.

We recorded 30 days of this period for analysis of overall player population and behaviour.

6.4.1 Population Demographics

Over the course of the 30 days recorded, the player base grew consistently between 5-15 players each day without major spikes or plateaus, as seen in figure 6.3. This reached the total amount of 289 users which indicates that most users that downloaded the mod went ahead and at least reached far enough to have their character registered in the server.

In figure 6.4 we can observe that there were some players that seemed to enjoy the game, some investing over 30 hours in the game in the span of 30 days. To better understand the playerbase we divided the various ids in several categories as can be seen in 6.5.

The darker colors represent ids that have no real player impact in the overall economy and gameplay, these being:

1. Eleven armies created as NPCs for players to challenge explained in section 4.4.4;
2. Forty-nine players that didn’t enter or play the mod correctly. These players are mainly players that entered the mod using previous campaigns and therefore would have an unfair advantage such as extra gold or troops; thankfully those players didn’t play more than 5 minutes of the mod and had little impact in the game;

3. Fifty-two players seemingly didn’t play the game and only got added to the server during character creation; therefore they count as having no engagement;

4. Most players, one hundred and four in total, are considered low engagement, meaning they had less than half an hour of play time;
5. Sixty-two players played between half an hour to two hours and are considered medium engagement;

6. Twenty-two players were considered high engagement, meaning those players played beyond two hours.

As we can see through the daily active users, figure 6.6, and the weekly active users, figure 6.7, the overall engagement had a declining factor over the course of the testing period. Which is to be expected as the initial burst of popularity due to the mod being a novelty worn down and only the most recurrent players maintained activity.

Figure 6.6: Daily Active Users

Figure 6.7: Weekly Active Users
In figure 6.8 we can see the overall player attendance over the course of the testing period. As we can see over 80% of the players didn’t login again after the first day, 10% seemed to give the mod more thought as they played again a second day and then 5% of the players clearly enjoyed the mod as they played for more than five days.

We can conclude that we couldn’t captivate most players enough to play the mod more than once. But the mod shows clear potential as some players became captivated enough to play recurrently and invest multiple hours into the game. We will discuss how to better improve our odds of retaining a bigger percentage of players further down the document.

Before heading to the high engagement players, we first wanted to understand what might be the causes underlying lack of commitment from players after the first day.

Like mentioned before, one of the underlying causes might be a way too harsh early progression
speed. The figure 6.9 depicts the last operational troop count (troops in party plus troops in garrison) and the respective playtime for the players in the low and medium engagement. As we can see, although a clear progression can be seen with the increase of play time, there are indeed some players that sit in the lower end of troop count even after some play time.

Since the distribution is relatively uniform we can not infer for sure that the underlying factor for player drop out is lack of troops and, the cooldown of wounded troops seems to be having the intended effect of slightly slowing down the player’s progression, so he doesn’t quickly exhaust the game. It also gives a strategic sense to the battle since players can’t mindlessly charge against far superior armies for a quick reward (since the loss of troops would result in a setback that would not make it worth it).

But it is clear that the system might be too harsh and some attenuation must be done for newer players until they get used to the gameplay and style.

6.4.2 High Engagement Players

We also need to analyse our most engaged players and understand what they enjoyed about the mod; to better understand the behaviour of our most engaged players we analysed their overall behaviour including gold earned, elo gained and the army power achieved over the course of 30 days. From now on we will show a selection of graphs, with the legend of player id, that represent the overall behaviour these players had during the testing period.

![Gold evolution over the 30 days](image)

In figures 6.10 and 6.11 we can observe the overall progress of gold and elo of these players. The elo in particular shows that the most active players had a consistent increase in elo throughout the testing period, but as we can see it often suffered minor drops which suggests that the elo was balanced enough and that the players didn’t necessarily have an easy road to the top.

We can also see in figures 6.12 and 6.13 that players heavily invested their gold in bigger and better
armies, some players even reaching troop counts of over a thousand troops and reaching considerably high army powers which suggests these troops weren’t low quality ones. We also have the figures 6.14 and 6.15 which allows us to also gauge on what is the players’ potential army power based on play time (objective hours invested rather than a progression over days of play time). As we can see the graphics maintain a similar proportion of progression which suggests that players invested a regular amount of time spread out throughout the days.

There is a slight outlier of the id number 130 that had a major spike in power and troop count. This player noticed a bug in the game that allowed to recruit thousands of troops for free, notified the developer and didn’t take advantage of the bug, as he didn’t use these extra troops.
Figure 6.13: Total troop count evolution over 30 days

Figure 6.14: Total army power evolution over play time

Figure 6.15: Total troop count evolution over play time
amount of troops clearly indicates that there was a need to compensate the troops being wounded after battle, and in turn made so players adapted into having large reserves to quickly replenish their troops, so they are able to fight more battles. Another reason could be that the players wished to have many options to customize their army.

Also, as it can be seen in the standing army in the party, figures 6.16 and 6.17, players strategy didn’t always resolve to having the biggest, most powerful army available. There were players that actively avoided having armies too large (one of such even constantly reduced his army to all but himself) to avoid being challenged and giving away too much money to the opponent. Other players also reduced their overall power to be able to challenge weaker opponents once the stronger ones were already used up for the day.

Some players also used the garrison system as it was originally conceived, as they would show considerably different army compositions from time to time. Which shows that the goal to allow the players to fully customize their armies from battle to battle was achieved.

In figure 6.18 we can observe a distinct pattern of a large ratio of victories for offensive battles and a large ratio of defeats for defensive battles. This tells us that players are good at predicting the possible outcomes of battle before deciding to challenge their opponent, also that even with the hardest difficulty set for the AI, the player factor is still a really large advantage in battle. The defeat ratio is also explained for the same factors, being unable to personally defend against a player will seriously hinder the chances of victory for the challenged party.

Figure 6.16: Party army power evolution over 30 days

In figure 6.18 we can observe a distinct pattern of a large ratio of victories for offensive battles and a large ratio of defeats for defensive battles. This tells us that players are good at predicting the possible outcomes of battle before deciding to challenge their opponent, also that even with the hardest difficulty set for the AI, the player factor is still a really large advantage in battle. The defeat ratio is also explained for the same factors, being unable to personally defend against a player will seriously hinder the chances of victory for the challenged party.
6.5 Open Survey

Along side the mod's distribution, a questionnaire was also made available for the players that were willing to give feedback, this questionnaire is available in appendix C. A total of 18 users answered the questionnaire which, for roughly the 300 players that played the game in the period of recording, amounts to around 6% of the players. There is also a big overlap between players that answered the questionnaire with the players that played for longer periods of time, which means there could be a slight bias in the overall results.

Of the 18 participants, 95% were male and 55% were under the age of 25, meaning a majority of
young players but still a fairly big presence of older players. Almost all players had over 50 hours in the
game and 44% of them had more than 200 hours, meaning the players that answered the survey were
fairly experienced in the game.

The survey was mostly composed of statements with a seven point Likert Scale to measure the level
of agreement the participants felt when reading them, from strongly disagree to strongly agree. The
statements were based on the “Game User Experience Satisfaction Scale” by Mikki Phan, Joseph R.
Keebler and Barbara S. Chaparro [11]. These statements were divided in two main sections, the usability
and enjoyment sections.

For the usability section, we wanted to measure if the mod was easy to pick up, if the interface
developed was intuitive and easy to understand, if the players could understand what to do without
requiring lengthy information or tutorials, and if the players understood the information given.

For the enjoyment section, we wanted to evaluate if the mod was fun to play, if the in-game loop
(challenging new players to battle) was fun, if the meta-game loop (upgrading the army after battle and
if having an army evolving as an addition were interesting) was fun. If the player felt proud when he won
and desired to try again when he lost.

These questions are displayed in figures B.15, B.16 and B.17, of the appendix B, with their respective
answer distributions.

After gathering all the players’ data, we applied the Cronbach’s Alpha test on our variables to evaluate
if there was enough correlation for the sections and see if we can group each section as a single variable
which would represent for one the usability and for the other the enjoyment. For the usability variable we
had an alpha of 0.842, which is a fairly acceptable value for merging the variables. For the enjoyment
variable we had an alpha of 0.818, which is also a fairly acceptable value for merging the variables.

The usability score hit a bit on the average side, it’s clear that most users understood how to operate
the mod but it’s also clear it could have some improvements. The enjoyment score has a higher average
which means that the players clearly enjoyed the mod.

We also had some open answer questions that allowed the players to write what they enjoyed the
most and the least in the mod and how the mod could be improved. Most answers were positive. The
biggest critic was that players quickly ran out of troops and that being on cooldown was boring (around
8 answers were along these lines). The most common praises were related to the army upgrading
feature, either fighting different armies made by other players or the customization of the battle itself
(around 6 answers were along these lines). Players also enjoyed that the mod was focused on battling a
lot. Most improvement suggestions were in essence an expansion of the mod. More maps, siege mode
(a mode that allows to siege enemy castles), getting loot after battles, having a unit marketplace that
allows to trade units, having a random matchmaking so that the player doesn’t always have to choose
an opponent.
6.6 Controlled User Tests

The *Nexus Mods* community is in its essence an already experienced community that desires a new experience of the game usually through the addition of mods available in the site. Although this helps us understand if the mod is desirable for the regular, more veteran player base of *Bannerlord*, it doesn’t say much about potential new players and how the mod could attract new potential players that don’t have the game yet.

To better evaluate this, we did 30 controlled test sessions with random people from or near our university. Our university is composed of a lot of engineers and software focused students so it is more likely to find people that enjoy gaming and are more prone to try new games. These tests were performed on the TagusPark campus over the course of a week and a half, and every participant was awarded a 5 euros *Steam Card* for the *Steam* platform as a reward for participating in the experience.

6.6.1 The Controlled Tests

The User Test Guide, available in the appendix D.1., is mainly focused on properly informing the tester on the overall test and ensuring his proper consent and participation on the test.

The test goes over on what would be the first 30 minutes of gameplay in the mod. It was therefore divided in 3 sections:

1. The Basics, where the user would get a grasp of the basic gameplay in *Bannerlord*. Players had around 15 minutes to enter the mod, recruit a few troops and go into the formation manager to get...
acclimated to the game’s combat system. Some players needed more time as there was no point in progressing if the basics weren’t at least semi understood.

2. The First Battle, where the player would challenge a small party specifically made for newer players composed of 5 looters (the weakest unit in the game). Then the player would use the money earned to either upgrade an existing unit or recruit new ones.

3. The Free Reign phase, where the player would get free reign on how to progress from then on. The player was tasked with fighting 5 battles and spending the gold at his own volition.

In general the tests went smoothly. Most users quickly understood their tasks and what to do, the biggest problem observed were players struggling to adapt to the game’s combat mechanics. Players seemed to enjoy the progression and getting their army larger. During the free reign phase, most followed the advice to challenge easier armies, since an equal or large army could result in a loss that could lead to a too big of a setback. There were some players that disregarded this advice and ended up losing most of their army which resulted in a harsher experience overall in the last task.

6.6.2 The Survey

After playing for about 30-45 minutes, depending on how quickly the testers picked up the pace, the testers were then asked to fill out a survey, available in the appendix D.2.

Of the 30 participants, 93% were male and 86% were between the ages of 18 and 25, meaning most testers were male students from the University of Lisboa. Over 86% of the participants had never played any Mount and Blade title and all of them have either never played the Bannerlord or have under 25 hours, meaning all the participants are fairly inexperienced in the game.

Much like the online survey we used the Likert Scale to measure the level of agreement the participants felt when reading a series of statements, from strongly disagree to strongly agree. The statements were also based on the “Game User Experience Satisfaction Scale” [11].

The statements for the usability and enjoyment sections were similar to the survey used for the online testers as we wanted to evaluate the same metrics. We did have to adapt some statements as they relied on game knowledge of the game which our participants wouldn’t have in just 30 minutes of gameplay. We also added a question in regards if the testers would play the mod again if they had the opportunity. These questions are displayed in figures B.19, B.18 and B.20, of the appendix B, with their respective answer distributions.

After gathering all the players’ data, we again applied the Cronbach’s Alpha test on our variables to evaluate if there was enough correlation for the sections and see if we can group each section as a single variable which would represent for one the usability and for the other the enjoyment. For the usability variable we had an alpha of 0.871, which is a fairly acceptable value for merging the variables.
For the enjoyment variable we had an alpha of 0.749, which achieves the minimum alpha value for the merging of variables to be acceptable, which is 0.700.

As we can see in figure 6.20 the average scores of both sections are higher than the online survey scores. This can be attributed to the lack of previous knowledge of the original game, which can skew testers opinions as some of the praise comes from the base game itself and not necessarily exclusively the mod. It also had the opposed reaction, many critics were actually aimed at Bannerlord rather than the mod itself.

For example, the horse mechanics had a lot of critics, with users often complaining that moving the horse was too complicated and that they didn’t understand how to control it. In general most players seemed to enjoy the game, even those that clearly stated that it was not their type of game. Players enjoyed the competitive nature of the game and some of the testers even asked me how to challenge their friends and they quickly developed a desire to out do each other.

Often times players lost track of time and how many battles they’ve done, some times doing one or two more battles than the original planned.

There were also a lot of good suggestions. One of such was so that players could train skills and troops in a training ground rather than just the battles, this could prove useful for players that had large armies in garrison. A tester suggested that there should be a quick action menu to give easier access to the most common interactions, for example a display with new players to challenge, troops to upgrade / recruit, etc. This means that the player doesn’t have to do repetitive actions in different interfaces.
6.7 Conclusion

From the overall analysis we can have a mixed conclusion of the mod. On one side, the reaction to the mod both in the open survey and the controlled tests survey was overwhelmingly positive and people claimed to both understand the mod and enjoy it. However, on the other side, the conversion rate, be it from page viewers to testing players and from testing players to regular players, was rather small.

So the for the positive results we have: a reasonable percentage of players invested multiple days and a lot of hours into the game, as well as answered the survey with fairly positive remarks and constructive critics and suggestions.

Players enjoyed the mod was focused around upgrading armies and battling a lot, for example: “I really like that the game mod is focused around a lot of battles.”, “Building army from scratch, Skill progression, Potential” and “The persistency and ‘pve’ gameplay aspect allowing for a casual play style.”. This means that the mod has potential and that a percentage of players enjoys the concept of the mod and that there is audience for this type of game mode.

The controlled tests also suggest the same, most testers answered the survey positively with high marks. A lot of players enjoyed the troop progression system as this was their answer to what they liked most about the mod: “Being able to improve my army”, “the possibility to upgrade troops individually”, “The evolution of your troops seems like a very good system” and “The upgrading troops feature”.

Knowing there is a target audience that enjoyed the concept of the mod and its implementation is key for any further work and improvement that could take place.

However we need to address some of the mod’s shortcomings. One of them being the player’s turnover after the first day. If the mod’s implementation by itself isn’t the sole factor, as some players clearly enjoyed the mod enough to play far beyond 30 hours, then most likely either the hook for new players is missing or the progression is too slow for some new players to keep them entertained enough for them to play again in the next day.

Since there isn’t a clear reason the best we can do is to present options that could attenuate this potential factor.

One potential solution would be a less harsh progression system for newer players. Like mentioned before, the wounded system is in place to keep players engaged for longer by preventing them from exhausting the content of the game too soon, but also to contribute with a strategy factor, or else there wouldn’t be any deterrent to just killing your army to earn more money which can break the immersion of a medieval combat game. We also can’t be sure that, without this mechanic, the players that played for over 30 hours would’ve dropped the game sooner due to the progression being far faster without wounded troops.

But this mechanic may have had the opposite effect to what was expected and by having it in place to newer players may have slowed down the progression too much to keep them hooked. So a possible
solution would be to give a 1 or 2 day wounded protection so the progress is easier in the beginning and only slows down when the player gets accustomed to the game. This however, as mentioned before, wouldn’t be possible to implement during the testing period as it would seriously affect the results and affect possible conclusions we would have for this thesis.

The game mode could also benefit of hook factors and daily motivations for the player to come back, like a daily login bonus, a daily challenge and even for a future project, seasonal content (changing the game content each season and resetting to persuade older players to come back and try the new content). This was however, already out of scope and serves only as a possible solution to the problem presented.

There were players that also raised the problem of a “barren zone” in the mod between the early game players and the stronger players, were the players making this jump would often feel lacking in equal opponents and therefore would seriously slow down in progress. A good solution for this problem would be to implement more NPC armies and with a bigger distribution in power, that served as a guarantee that regardless of your stage of progression, you always have some armies available to challenge and help you progress.
## Conclusions and Future Work

### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Conclusions</td>
<td>89</td>
</tr>
<tr>
<td>7.2 System Limitations and Future Work</td>
<td>90</td>
</tr>
</tbody>
</table>
7.1 Conclusions

This document describes the design, implementation and testing process for a new game mode for the game *Mount & Blade II: Bannerlord*, with the goal of understanding if a multiplayer experience with meta-game progression associated with it could be a good combination to better the engagement and enjoyability of players.

During these months a prototype was developed that successively implemented these features and managed to form a full fledged prototype, capable of being freely used by any person that desired to test the mod. From the feedback received, as well as analysis to the data obtained, we can consider this project succeeded in its original goal of creating an enjoyable multiplayer experience with persistent elements, but given the failure to win over most users that decided to download the mod we can say that the project could certainly use a lot of improvement.

Players really enjoyed the concept and even newer players that openly said the game wasn’t their style, had fun playing the game and would like to try again. In addition, most of the recurrent players of the mod enjoyed the mod not only for what it was, but also for what it could be if the project were to be further developed. A lot of suggestions were given to further expand the mod, such as siege battles, having some NPC missions that allowed the player to do side missions to progress without requiring challenging new players, a form of CO-OP, were you could fight battles of 2 versus 2 or more, etc. In essence, players saw much potential in the concept which is essential if this project were to be continued.

We also need to address some shortcomings. The mod seems to lack a good hook factor as much of the players don’t stick to the mod after the first day.

One of the biggest complaints were relative to the wounded troop system, as the players feel it hinders them too much of playing the game; but, as we saw, players found a way to counter this problem by having large reserves of troops, which means the problem is solvable by the players, but only at later stages. To help attenuate this problem and keep players more engaged on their first day, a good possible solution is to give a “new player” protection that stops troops from being wounded for the first 1 to 2 days.

A second possible solution to be added in addition to the previous one, is to add more reasons to motivate players to login regularly, such as login rewards or daily quests / challenges. Adding more content to the mod tied to consistent gameplay throughout the days should help trigger the desire of players to open the game again and in turn to play the game a little longer.

Another improvement point is more NPC armies with a bigger distribution of power. A few players complained that at certain stages the game felt barren and lacking players to challenge, as they were too strong for the majority of players but still too far away of the top strongest players, and they felt the grind in this “barren zone” was a bit boring to an otherwise good game.
7.2 System Limitations and Future Work

The General Lords’ system currently has two big limitations for its future, but both are manageable with work and time.

The first is related to Bannerlord itself, as it is a game still in development and receives monthly updates. These updates sometime break the mod, or some of its components, which requires intervention by the modder in order for the mod to become operational again. This could be mitigated if some systems were improved in order to depend less on the base version and more on the mod’s implementation itself. It could also be an opportunity to make some features server side, rather than client side which can compromise the security and well functioning of the mod.

This leads us to the second big limitation, which is the server itself. Currently the server serves only as a storage and sharing service that allows multiple players to connect and see other potential adversaries. It has some complex interactions, but in general it was created to serve the purpose of supporting the mod and not the other way around. A full fledged game should have a robust server that doesn’t rely as heavily on the correct functioning of the client side. This would help avoid potential cheaters, hacking attacks and bugs.

In short, below is a small list with the most important future work in order:

- **Further Develop the Server:** The first key feature that needs to be worked on is the server. As of right now, it serves only as a fetch and storage server that allows communication between players and simulates proper asynchronous multiplayer. However it needs to be improved as it is rather inefficient and can’t support a much larger player base (as it starts to increase the overall lag of the gameplay). It also needs proper authentication system and verifications against cheating.

- **Daily Features:** The mod would need daily login rewards to help kick start the players’ progress each day, as well as motivate them to show frequently to play. Some extra content such as daily quests or challenges would also freshen up the content and help keep players engaged for longer.

- **Seasons:** In addition to daily features, seasonal content should help revitalize the game and bring back older players to try the game again.

- **Perks:** The perk system wasn’t properly implemented due to limitations on the system during development, although it might be possible to, with more time, refactor the system into being compatible with the server and sharing between games.

- **Recording battles:** Defensive battles are kind of overlooked as the player has no way to understand what really happened in the battle and if their defensive formation worked. Therefore adding recordings of battles for players to review what happened in battle could motivate players to invest in defensive tactics and to try new strategies.
Bibliography


Game References

[21] Begginer guide and upgrading your tank, https://www.youtube.com/watch?v=2S0y9UniRt8&ab_channel=DauntlessTanker
[27] Phasmophobia, https://store.steampowered.com/app/739630/Phasmophobia/


[34] Smith, A.: Hands on: Total war - arena (December 2014), https://www.rockpapershotgun.com/2014/12/15/hands-on-total-war-arena/
A.1 Battle Plans

A.1.1 Phases

The battle plan divides the battle into 5 different phases, where he can specify what the troops should do in each stage:

- **Position:** The position phase is the starting phase that automatically instructs the player army to move onto the desired positions and formations before the battle commences. These positions and formations are based on the chosen formation by the player.

- **Prepare:** When the battle starts the player and his army is still too far from the enemy to properly engage in any kind of battle. So in the prepare phase the player gets to decide how his army will approach the enemy.

- **Ranged:** After some distance has been shortened the armies will eventually be in range of missile units such as archers or crossbow men. The player can decide how his troops approach the enemy.
whilst archer fire is being traded with both sides.

- **Engaged**: In the engagement phase, troops should be in imminent clash with each other and melee fighting is about to ensue. It's up to the player how and what troops will engage in melee combat.

- **Winning / Losing**: After a few minutes of battle, a winner is sure to come out. So the player has the ability to select how his troops should handle the finishing stages of the battle, allowing him to specify what to do whether they are winning or losing.

### A.1.2 Orders

In each individual phase, the way that commands are decided is by assigning each formation (infantry, archers, cavalry, etc.) what to do using the 8 available orders:

- **Advance**: The designated formation is commanded to advance at a steady pace forward towards the enemy.

- **Cautious Advance**: The designated formation will approach slowly and steady towards the enemy.

- **Charge**: The designated formation will charge at full speed towards the enemy.

- **Flank**: The designated formation will try to position itself at either flank of the enemy.

- **Protect Flank**: The designated formation will try to position itself at either flank of the other allied units.

- **Hold Position**: The designated formation will halt and hold their current position.

- **Hide Behind**: The designated unit will try to move behind the main allied body of troops.

- **Skirmish**: Missile units in the designated formation will try to shoot at the enemy whilst trying to maintain a minimum safety distance.

### A.2 Mod Description

The description was composed of the following:

**What is General Lords?**

General Lords is a semi-multiplayer experience where you get to challenge other player's armies to see who has the best one and the best strategies. You get to create your army and customize it to
your liking using the six factions’ troop trees and further improve it using the new developed Battle plan and Formations that let you designate automatic troop behaviors and automatic formation deployment in battle. Earn gold by fighting battles and killing your enemies! After each battle, use it to upgrade your troops and recruit more to kill even more enemies! In a thrilling cycle of war where you get to climb the ladder to become the best General Lord!

This mod is the final proof of concept of a university thesis on game development and game design.

Even though it has been thoroughly tested, the mod might still contain some bugs. If you want to help clear them you can play with the mod “Better Exception Window” and send a print screen of any crash to the mod creator. If you run into any trouble or if you want to get in touch with the creator or other members of the community then join our discord server and choose the blue color to be redirected to the General Lords’ discord rooms where the creator publishes announcements and patch notes, and where you can chat.

**Warning**

The mod is meant to be played in English! The server works with English information and using a different language on the game could cause problems!

Each save file created and used in this mod is expected to be played whilst using the mod. So no guarantees are given if you try to use the save file without the mod activated. The mod also doesn’t envision the use of other mods, so no compatibility is ensured!

**How to start a game**

After the game is installed, start a new campaign normally. This campaign will be your General Lords’ campaign from there on, so please refrain from using it outside of the mod.

Create your desired character and select your culture of choice (this will affect your early choices for available troops and items).

Afterwards most information is available when you hover the title of the mod on top of your screen. We strongly recommend for new players to spend their early money in recruits, as fighting alone could prove too difficult! Money is acquired by killing troops, so fighting alone could prove hard to earn more money.

**I need your help!**

This mod is a proof of concept that aims to evaluate the enjoyability of the game mode. It was made by only one person and with time restraints, therefore I would like your help by playing the mod honestly and avoiding cheating or doing unintended actions outside of the mod. Please only play saved files started whilst using the mod and only play them with the mod active.

If you do play the mod, it would be a big help if you answer this questionnaire. Your feedback is crucial for the future development of this mod!

**Installation**
We recommend that you install the files manually, by copying the General Lords’ module into the game’s modules.

**More Mods**
Want to try even more mods? Check out the ones created by my colleagues!

- Cunning Lords
- Friendly Lords
- Plot Lords
- New Lords
Figure B.1: Post in the r/Bannerlord page
Figure B.2: Post in the r/MB2Bannerlord page

Figure B.3: Post in the Talewords Bannerlord Forum

Figure B.4: Reply to the post in r/Bannerlord page
Figure B.5: Discord’s feedback 1 to the mod

Figure B.6: Discord’s feedback 2 to the mod

Figure B.7: Discord’s feedback 3 to the mod

Figure B.8: Discord’s feedback 4 to the mod

Figure B.9: Discord’s feedback 5 to the mod

Figure B.10: Discord’s feedback 6 to the mod

Figure B.11: Nexus Mods’ feedback 1 to the mod
Figure B.12: Nexus Mods’ feedback 2 to the mod

Figure B.13: General Lords being trending in Nexus Mods

Figure B.14: General Lords being on the most downloaded page of Nexus Mods
Figure B.15: Online Survey Usability Likert Answers Distribution

- I feel the mod provides me the necessary information to accomplish a goal within the mod: 3 Strongly Agree, 3 Agree, 2 Somewhat Agree, 1 Neither Agree nor Disagree, 7 Neither Disagree nor Agree, 3 Somewhat Disagree, 6 Disagree, 6 Strongly Disagree.
- I always know what to do next when I finish an action in the mod: 1 Strongly Agree, 2 Agree, 3 Somewhat Agree, 6 Neither Agree nor Disagree, 6 Neither Disagree nor Agree, 2 Somewhat Disagree, 2 Disagree, 6 Strongly Disagree.
- I find the mod's menu user friendly: 1 Strongly Agree, 3 Agree, 3 Somewhat Agree, 12 Neither Agree nor Disagree, 2 Somewhat Disagree, 2 Disagree, 6 Strongly Disagree.
- I always know how to achieve my goals/objectives when using this mod: 1 Strongly Agree, 1 Agree, 8 Somewhat Agree, 3 Neither Agree nor Disagree, 6 Neither Disagree nor Agree, 5 Somewhat Disagree, 2 Disagree, 2 Strongly Disagree.
- I find the mod's main screen easy to read and interpret: 1 Strongly Agree, 4 Agree, 4 Somewhat Agree, 11 Neither Agree nor Disagree, 2 Somewhat Disagree, 2 Disagree, 2 Strongly Disagree.
- I find the mod's interface easy to navigate: 1 Strongly Agree, 3 Agree, 3 Somewhat Agree, 11 Neither Agree nor Disagree, 3 Somewhat Disagree, 2 Disagree, 3 Strongly Disagree.

Figure B.16: Online Survey Enjoyment Likert Answers Distribution

- I felt immersed in the game when playing this mod: 2 Strongly Agree, 3 Agree, 4 Somewhat Agree, 5 Neither Agree nor Disagree, 8 Neither Disagree nor Agree, 4 Somewhat Disagree, 1 Disagree, 1 Strongly Disagree.
- You feel incentivized to play the game frequently to keep your army on par and not fall behind: 4 Strongly Agree, 5 Agree, 5 Neither Agree nor Disagree, 2 Neither Disagree nor Agree, 6 Somewhat Disagree, 2 Disagree, 1 Strongly Disagree.
- After concluding all battles of the day, you feel exited to play the game again tomorrow or later: 2 Strongly Agree, 3 Agree, 5 Somewhat Agree, 5 Neither Agree nor Disagree, 6 Neither Disagree nor Agree, 2 Somewhat Disagree, 2 Disagree, 2 Strongly Disagree.
- When you lose a battle you feel a desire to improve your army and try again: 3 Strongly Agree, 2 Agree, 5 Somewhat Agree, 4 Neither Agree nor Disagree, 4 Neither Disagree nor Agree, 2 Somewhat Disagree, 4 Disagree, 4 Strongly Disagree.
- I feel proud when I win a hard battle using my personal army: 1 Strongly Agree, 1 Agree, 3 Somewhat Agree, 10 Neither Agree nor Disagree, 3 Neither Disagree nor Agree, 10 Somewhat Disagree, 10 Disagree, 10 Strongly Disagree.
- An evolving personal army is a fun addition to a multiplayer game: 8 Strongly Agree, 10 Agree, 10 Neither Agree nor Disagree, 10 Somewhat Disagree, 10 Disagree, 10 Strongly Disagree.
- It's important to upgrade my army: 1 Strongly Agree, 2 Agree, 3 Somewhat Agree, 12 Neither Agree nor Disagree, 12 Neither Disagree nor Agree, 12 Somewhat Disagree, 12 Disagree, 12 Strongly Disagree.
- I enjoy upgrading my army after a victory: 10 Strongly Agree, 10 Agree, 7 Neither Agree nor Disagree, 8 Neither Disagree nor Agree, 8 Somewhat Disagree, 8 Disagree, 8 Strongly Disagree.
- It's fun to challenge new armies: 5 Strongly Agree, 7 Agree, 7 Neither Agree nor Disagree, 8 Neither Disagree nor Agree, 7 Somewhat Disagree, 7 Disagree, 7 Strongly Disagree.
Figure B.17: Online Survey Boxplots

Figure B.18: Control Test Usability Likert Answers Distribution
### Figure B.19: Control Test Enjoyment Likert Answers Distribution

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Somewhat Agree</th>
<th>Neutral</th>
<th>Somewhat Disagree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>You would like to try the mod again if you had the opportunity to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I felt immersed in the game when playing this mod</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When you lose a battle you feel a desire to improve your army and try again</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I feel proud when I win a hard battle using my personal army</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The concept of an evolving personal army is a fun addition to a multiplayer game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It's important to upgrade my army</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I enjoy upgrading my army after a victory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It's fun to challenge new armies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Figure B.20: Control Test Boxplots

![Control Test Survey Likert Scale Boxplots](image-url)

**Questions**

- Desire to Improve Army
- Easy/Achieving Goals
- Easy/Intuitive Navigation
- Easy/Interpretable Interface
- Evolving Army/Size
- Fun/Challenging Armour
- Immersed
- Information Provided
- Manual User-Friendly
- Next Action
- Proud/Winning Battles
- Try Again
- Upgrading Army/After Battle
The following are the questions asked in our open testing survey designated for players that found the mod either through recommendation or from the Nexus Mods. The survey was designed and performed on the Google Docs platform. Mandatory questions have an asterisk at the end.

General Lords’ Questionnaire

Thank you so much for testing and playing the mod “General Lords”! This mod represents a preview of what a multiplayer with army progression would feel like and the thesis around it aims to evaluate if such a game mode would be fun and enjoyable by most players. Therefore this questionnaire serves as a way to gauge if the players found the game mode fun and if they would play again should they have the opportunity.

This questionnaire should take no more than 10 minutes to complete and it is completely anonymous if you choose. Any data collected only being used in the scope of this project By pressing “Next” you are agreeing to share with us any data you choose to reveal in your various answers.

Demographics

1. Gender * [Male; Female; Prefer Not To Say; Another Option]
2. What is your age? * [Intervals]

3. How many hours have you played "Mount & Blade II: Bannerlord"? * [Intervals]

4. Have you played other Action-RPG games? * [Yes/No]

5. If so, which games have you played? (Separate the names of the game with a comma eg: name1, name2) [Short Answer]

6. Do you often use mods or do you prefer to use only the native game?

7. How many hours have you played the “General Lords” mod? * [Intervals]

**Mod Usability**

1. I find the mod’s interface easy to navigate * [Strongly disagree 1 - 7 Strongly agree]

2. I find the mod’s main screen easy to read and interpret * [Strongly disagree 1 - 7 Strongly agree]

3. I always know how to achieve my goals/objectives when using this mod * [Strongly disagree 1 - 7 Strongly agree]

4. I find the mod’s menus user friendly * [Strongly disagree 1 - 7 Strongly agree]

5. I always know what to do next when I finish an action in the mod * [Strongly disagree 1 - 7 Strongly agree]

6. I feel the mod provides me the necessary information to accomplish a goal within the mod * [Strongly disagree 1 - 7 Strongly agree]

**Mod Enjoyment**

1. It’s fun to challenge new armies * [Strongly disagree 1 - 7 Strongly agree]

2. I enjoy upgrading my army after a victory * [Strongly disagree 1 - 7 Strongly agree]

3. It’s important to upgrade my army * [Strongly disagree 1 - 7 Strongly agree]

4. An evolving personal army is a fun addition to a multiplayer game * [Strongly disagree 1 - 7 Strongly agree]

5. I feel proud when I win a hard battle using my personal army * [Strongly disagree 1 - 7 Strongly agree]
6. When you lose a battle you feel a desire to improve your army and try again * [Strongly disagree 1 - 7 Strongly agree]

7. After concluding all battles of the day, you feel exited to play the game again tomorrow or later * [Strongly disagree 1 - 7 Strongly agree]

8. You feel incentivized to play the game frequently to keep your army on par and not fall behind * [Strongly disagree 1 - 7 Strongly agree]

9. I felt immersed in the game when playing this mod * [Strongly disagree 1 - 7 Strongly agree]

Open Ended Questions

1. What was your first impression of this mod?

2. What did you dislike about the mod? Why?

3. What did you like most about the mod? Why?

4. What feature did you wish was present in the mod?
D.1 User Test Guide

D.1.1 General Lords User Test and Consent Form

General Lords User Test and Consent Form

Dear participant,

We are conducting a study that aims to evaluate the overall enjoyability of a mod, called General Lords, which introduces a new game mode developed for the game Mount and Blade II: Bannerlords.

The game mode essentially consists of a semi-multiplayer game where you get to develop your army and challenge new different players to show who has the best army created.

In this session you will be playing 40 to 45 mins of the game mode and then answer a quick questionnaire regarding your experience playing the mod.

Your participation is completely voluntary and you are free to leave or stop participating in the study at any moment of the session.

We would like to request you read the following consent form and, if you consent to participate in the
study with us after reading the terms below, to sign the document in the indicated spot.

Thanks for participating!

D.1.2  Consent Form

Consent Form

1 - I read and understood the meaning of this study. I had the opportunity to make questions when needed and receive the respective answers.

2 - I understand that the participation in this study is voluntary and I can quit at any moment, without requiring justification. If such a thing occurs, there won’t be any consequences and my data and information will be erased and won’t be used in the study.

3 - I consent my data to be used in the research, study and sharing of results in future scientific publications or conferences concerning the project, by the researchers of this project.

4 - I understand that the data retrieved in this study will be used like mentioned before.

5 - With agreement with the previously read, I consent to my participation in this study and accept its conditions.

Participant Date

----------------------------------------

Interviewer:
Marcos Pêgo marcosppego@gmail.com

D.1.3  General Lords User Test Guide

General Lords user test guide

You will now have a small sequence of tasks to test how the users would fare when interacting with the mod. These tasks are the following:

Task 1 Basics (15 minutes) - Open the game. Start a new campaign. Create and customize your character. Conclude the character creation process and enter the main game. Recruit 5 to 10 recruits. Open the Formation Manager. Learn the basics of movement and test out the battle. Conclude by killing the sole soldier on the other side of the screen.

Task 2 First Battle (5 minutes) - Open the match selection screen. Find and challenge the tiny looters party. Conclude the battle.

Task 3 Upgrading your army and character (5 minutes) - Upgrade a few troops (if no upgrades are available, recruit a few recruits). Upgrade your character’s focus and perks.
Task 4 Battle your opponents (15 minutes) - Scout your opponents to see winnable / similar matches. Win or fight 5 battles. Use the gold earned from these battles however you please.

D.2 User Test Survey

The following are the questions asked in our controlled test designated for the participants after they concluded the 30 minutes session. The survey was designed and performed on the Google Docs platform. Mandatory questions have an asterisk at the end

**General Lords’ Questionnaire**

Thank you so much for testing and playing the mod “General Lords”! This mod represents a preview of what a multiplayer with army progression would feel like and the thesis around it aims to evaluate if such a game mode would be fun and enjoyable by most players. Therefore this questionnaire serves as a way to gauge if the players found the game mode fun and if they would play again should they have the opportunity.

We would also like to remind you that: - participation is voluntary and you can withdraw at any time; - at no point throughout the study will you be identifiable, and individual results will not be disclosed; - there are no physical or psychological hazards in your participation; - you are free to ask any inquiry about the experiment at any time, and you can also join the project’s discord server if you have any further questions.

**Demographics**

1. Gender * [Male; Female; Prefer Not To Say; Another Option]
2. What is your age? * [Intervals]
3. Have you ever played any Mount and Blade game? * [Yes/No]
4. How many hours have you played “Mount & Blade II: Bannerlord”? * [Intervals]

**Mod Usability**

1. I find the mod's interface easy to navigate * [Strongly disagree 1 - 7 Strongly agree]
2. I find the mod's main screen easy to read and interpret * [Strongly disagree 1 - 7 Strongly agree]
3. I always know how to achieve my goals/objectives when using this mod * [Strongly disagree 1 - 7 Strongly agree]
4. I find the mod's menus user friendly * [Strongly disagree 1 - 7 Strongly agree]
5. I always know what to do next when I finish an action in the mod * [Strongly disagree 1 - 7 Strongly agree]

6. I feel the mod provides me the necessary information to accomplish a goal within the mod * [Strongly disagree 1 - 7 Strongly agree]

Mod Enjoyment

1. It’s fun to challenge new armies * [Strongly disagree 1 - 7 Strongly agree]

2. I enjoy upgrading my army after a victory * [Strongly disagree 1 - 7 Strongly agree]

3. It’s important to upgrade my army * [Strongly disagree 1 - 7 Strongly agree]

4. The concept of an evolving personal army is a fun addition to a multiplayer game * [Strongly disagree 1 - 7 Strongly agree]

5. I feel proud when I win a hard battle using my personal army * [Strongly disagree 1 - 7 Strongly agree]

6. When you lose a battle you feel a desire to improve your army and try again * [Strongly disagree 1 - 7 Strongly agree]

7. I felt immersed in the game when playing this mod * [Strongly disagree 1 - 7 Strongly agree]

8. You would like to try the mod again if you had the opportunity to * [Strongly disagree 1 - 7 Strongly agree]

Open Ended Questions

1. What was your first impression of this mod?

2. What did you dislike about the mod? Why?

3. What did you like most about the mod? Why?

4. If you could change or add anything to the mod, what would it be?