A Procedural Quest Generator for M&B II: Bannerlord - Extended Abstract

Marco Cabral, 87680
Instituto Superior Técnico, 2744-016 Porto Salvo, Portugal
marco.cabral@tecnico.ulisboa.pt
https://tecnico.ulisboa.pt/en/

Abstract. We propose for a procedural quest generator for the game Mount & Blade II: Bannerlord. Although Bannerlord is a game that already has a questing system, the quest it presents are repetitive, leading to a gameplay experience with a tendency to become dull. With the procedural quest generator, we aimed to implement a questing system that can create an almost endless amount of quests that seem different from one another, providing the player with a prolonged and varied experience. This generator can also allow for a lower cost of production when creating quests since a game developer wouldn’t have to spend so much time creating every single individual quest. The implemented system was based on a group of grammar rules that were inspired by the work developed by Doran and Parberry for quest generation. We will also analyze the different state-of-the-art engines made towards interactive storytelling to make sure the developed quests are coherent and believable in the storyline of the game. To properly evaluate the developed system, we conducted user tests and published the resulting mod online. These tests, which evaluated the quality and enjoyment of the quests, showed that our system was able to generate quests that could be enjoyed by players and presented a quality comparable to the Bannerlord quests.

Keywords: Mount & Blade II: Bannerlord · Procedural Generation · Quests

1 Introduction

Gaming has been an ever-increasing market, with each year getting more and more players. With this increase in demand, game companies have been spending even more time and funds creating their games than before, however, the consumption of the content present in the games is much faster.

This situation can be especially verified in the Open World Role Playing Games(RPGs) genre of games. This genre gives the player the possibility to play as a character or more from the game world that must complete quests, that might be connected or not, and follow the general story to reach an end goal and the conclusion of the storyline. The players can also develop their characters through the several decisions presented to them.
These genres are known for being content-heavy, with lots of non-playable characters (NPC's) together with a vast open-world to explore. All of this is used to create both the main quests and side quests of the game. The problem starts when a player completes all the quests and accomplishments (like collectibles) available to him, since it makes the replay value of the game decrease substantially. This makes these genres of games some of the perfect candidates to implement a Procedural Quest Generation System.

With the use of automatically generated quests, the players can both enjoy an almost endless game experience, and also replay the game without repeating the same quests over and over again. This also allows the developers to spend less time writing countless quests and more time refining other areas of the game. With this automatic generation of content, we can furthermore learn the player gameplay preferences, desires, abilities, and such and have the content presented to them adapted, possibly giving them a more personalized experience.

1.1 Problem Definition and Thesis Goals

One example of the kind of game talked about above is Mount & Blade II: Bannerlord (commonly referred to as only Bannerlord). Bannerlord is an Open-World, RPG type of game part of the Mount & Blade series with a medieval setting, where the player moves its character through the virtual world, completes quests (or missions) and conquers towns and villages to achieve its goal. At the time of writing this thesis, it possesses a somewhat limited and repetitive questing system. Each game day, a random number of quests are randomly chosen and applied to the NPC’s that can receive them, for the player to complete or ignore. Although this allows for an endless experience for the player, since even though the main quest ends, the side quests are continuously assigned, it quickly becomes repetitive and tedious. This is a situation that might however change or improve, since the game is still in development.

So one of the goals here is to have a system that creates somewhat interesting quests, that present more variation in both type and structure, through the use of an automated algorithm that the game can then read and transcribe to in-game quests in order to lower the production costs of the quests while maintaining a similar quality to the quests that are manually created. To accomplish this, we took a look at procedural quest generation systems to better understand how we can apply this concept to the game Bannerlord.

We analyzed other procedural quest generation systems, and we inspired ourselves in the work done by Jonathan Doran and Ian Parberry [1] with the modifications applied by António Machado [4]. This work is a quest system based on RPG games that presents a lot of quest variation, and we made an adaptation of it to apply in the game Mount & Blade II: Bannerlord.

2 Related Work

The focus of this work will be procedural quest generation. Procedural generation is a means of creating data algorithmically rather than manually, using a
combination of human-generated materials and algorithms, as well as computer-generated randomness and processing power. Let’s now look at the work that had the most influence on this thesis.

2.1 Doran and Parberry’s Prototype Quest Generator

According to Doran and Parberry’s work on a prototype quest generator derived from the analysis of quests from MMORPGs (Massively Multiplayer Online Role-Playing Game) [1] a procedural quest generation system has the potential to increase the variability and replayability of games, leading to an increase in player interest in these games, since there is never a moment where the player has experienced and completed everything in the game. Through the analyzes of almost 3000 quests from several games, Doran and Parberry concluded that even human-authored quests display significantly less structural variety than one might expect. By exploiting this common structure between the quests, they were able to show a prototype system that procedurally generates random quests appropriate to be used in RPGs.

The quests analyzed by Doran and Parberry fit into 9 different motivations: Knowledge, Comfort, Reputation, Serenity, Protection, Conquest, Wealth, Ability, and Equipment. These motivations correspond to the most important concern an NPC has, and the goal of a quest created by the prototype system would be to resolve this concern. The motivations have associated to them strategies that, through the use of actions, dictate how a quest should be followed in order to reach the goal.

We can represent the actions as an infinite set of trees, where the leaves are atomic actions that the player can carry out and the internal nodes represent tasks that need to be achieved along the way.

At the end of their work, Doran and Parberry, stated that, although they had built a functional prototype quest generator based on NPC motivations, further work would be needed to demonstrate its ability to generate quests that are as good as quest hand made by developers.

2.2 Towards a Procedurally Generated Experience: A Structural Analysis of Quests

One example of further work on Doran and Parberry quest generator is the work done by António Machado [4]. Here, António Machado, made an analysis of the main quests from the award-winning single-player RPG game ‘The Witcher 3 - The Wild Hunt’ [10] and implemented a grammar that extends what was previously presented by Doran and Parberry.

This grammar extension is, as claimed by the author, more expressive than the previous, which is something we agree on. This was accomplished by representing every main quest from ‘The Witcher 3’ in the same manner as Doran and Parberry had defined.

What was noticed while representing the quests was that, even though the NPC’s that gave out the quests had the same or similar motivations as the ones
defined by Doran and Parberry, the actions proved to have limitations making it impossible to fully describe the quests.

So a bigger analysis of the quests was conducted, where, while building the quest line trees as described by Doran and Parberry, the necessary changes were added. These changes were then attached to the previously built tables by Doran and Parberry.

António Machado implemented this work in the game Conan Exiles [3] developing a procedural quest generation system into a game that lacked any kind of questing system.

3 The Game - Mount & Blade II: Bannerlord

Bannerlord [7] is the fourth game of the Mount & Blade series, released and developed by TaleWorlds Entertainment. It's an Open-World, Action, RPG (Role-playing game), Simulation, Strategy game (according to Steam, which is an online game platform) that was announced in 2012 and had its early access version released on March 30, 2020.

The game features both a single player and a multiplayer mode, but we implemented our mod for single-player. Looking at the single-player quest system in more detail, the quests can be handed out by either an NPC living in a town, castle or village or an NPC roaming around the virtual world with its army. This is represented with a blue exclamation mark next to their avatar. When talking with these NPC’s with quests they have extra dialog, in comparison with the other NPC’s, so they can explain the quest to the player. After accepting a quest it appears on the quest management screen with the title of the quest, a small description, our current progress and step, and how many days there are left to complete it. There are currently (29 October 2021) over 25 different quests, all with a different goal and a different way to achieve said goal. In the examples below, we accepted a quest from an NPC that asked the player to train some soldiers.

4 Modding in Bannelord

To implement the questing system, we made use of modding 1. Bannerlord is relatively easy to start modding and there are plenty of tutorials on how to do it. To distribute our mod, we only need to share the Mod folder with other players. One of the main challenges when designing the mod came from the fact that, since it’s a recent game (still in early access at the time of writing this thesis), there is no available documentation of the game, and we had to make use of our IDE (Integrated development environment) and trial and error to know what methods we had available to implement the quest system.

1 'an alteration by players or fans of a video game that changes one or more aspects of a video game, such as how it looks or behaves’ [6]
5 Implementation - A General Overview

The goal is to implement a procedural quest generation system for Bannerlord, a game that already has a questing system. With this new system, we aim to improve Bannerlord by providing quests that are less repetitive and have a different more varied structure, while at the same time reducing the effort needed to make every single individual quest. With the new system, we can produce an almost infinite number of different quests. To accomplish this, three modules have been developed: the Generator module, the Instantiator module and the Monitor module.

The Generator module is responsible for building a behavior tree-like quest with placeholder names for the world objects (such as ”place1”, ”npc1”, ”item1”) similarly to what was proposed by Ian Parberry and Jonathon Doran with the modifications made by António Machado. The generated quests present a lot of variation in both quest type and structure. The Motivations selected for the quest are chosen at random, this allows the next module to have plenty of different quests with different motivations to assigned to the correct NPC’s.

The Instantiator module, as said before, is responsible for taking the quests generated by the previous module and assigning them to NPC’s with the same Motivation. This is module is also responsible for assigning the Motivations to the NPC’s, that are randomly assigned depending on the type of NPC (i.e.; if it’s a noble NPC it can have Reputation type quests, but a farmer NPC can’t). After assigning a quest to a NPC, it then takes the placeholder world objects of each action and replaces them with world objects that currently exist, this is, it replaces ”place1” with a settlement, ”npc1” with an NPC and ”item1” with an item.

Finally, the Monitor module starts when the player accepts the quest. Its job is to update the quest logs that appear on the quest screen by monitoring the player progress throughout the quest. This is done by listening to the several events the game launches every time certain actions happen. It also assigns dialogue to the NPC’s so that the player can progress in the quest.

6 Evaluation and Results

To evaluate the quality and enjoyment of the mod we made, at first we decided to publicly publish the mod and later to perform user tests. Through this we were able to determine if the generator we made generated more varied quests with a more diverse structure and if the players got the idea that the quests were made by humans. This will allow seeing if we can reach our goal of generating quests in an automatic way that are considered as good as the quests from Bannerlord.

6.1 Metrics

To know how our quests compared with the Bannerlord quests, we evaluated two different metrics, Quest Quality and Quest enjoyment.
To evaluate Quest Quality, we created a set of questions that would help us better understand what the player thought. These were, Quest Length, Quest Cohesion, Quest Step Clarity, Quest Description Clarity, Quest Dialogue Immersion and Quests made by humans. To evaluate Quest Enjoyment we used an adaptation of the subscale "enjoyment" from GUESS; Phan et al. 2016 [5], that specifically targeted the quests, so that we could know if the players enjoyed the quests. The questions were: "I think the quests are fun", "I feel bored while completing the quests", "If given the chance, I want to complete more quests", "I enjoy completing the quests" and "I feel like the quests are repetitive".

6.2 Mod Publication

The mod was published on the most popular mod website for Bannerlord mods: Nexus mods [8]. It was published on the 22nd August 2021 [9] and in 60 days it reached over 2600 unique downloads, over 19700 views and over 35 endorsements (a way for users to show their appreciation for a mod). One of the goals with the publishing of the mod online was to have players with experience in the game to answer a questionnaire. Even though we had many people downloading the mod, we, unfortunately, only ended up with 8 answers to the questionnaire, meaning that they are not statistically relevant enough for us to analyze, so we had to resort to in-person user tests to have results. This lack of answers might be due to the fact that there was no incentive for people to answer the questionnaire.

6.3 User Tests

Due to not having enough answers in the online questionnaire, we decided to perform user tests. These were done in the span of two weeks, the participants were part of the Instituto Superior Técnico - TagusPark community, which is also where the tests were conducted. Each participant was given a 5€ gift card to use on the steam platform as incentive to participate. Accompanying the user tests, we had the players fill out a questionnaire. This test had a duration of between 30 and 40 minutes. The questionnaire consisted of a Demographics part and then two sections both used to evaluate the quest quality and quest enjoyment, one section for the first two quests and the other section for the last two quests (it could either be: First two quests → Bannerlord quests, Last two quests → Plot Lords quests or First two quests → Plot Lords quests, Last two quests → Bannerlord quests). The players were also accompanied during the test in case they had any questions regarding the game since, for most of them, it was their first time playing Bannerlord.

The user Tests were conducted as shown in Figure 1.
6.4 Results

Regarding demographics. We had a total of 31 participants, where 87% were male and 13% were female. The overall age of our participants was between 18-25 years old, and a bit over 60% of them played games on a daily basis. The rest of them said they played weekly (13%), monthly (13%), once in 6 months (3%), once a year (3%) and either less than once a year or never (6%). Only one participant had played Bannerlord before, and only 3 participants played M&B Warband before. However, over 87% claimed to have played other RPGs/Open World games before.

**Quest Quality** A Cronbach Alpha test was conducted to measure the internal consistency of the questions. It was done for both the questions related to the Bannerlord quests and to the Plot Lord quests. Bannerlord quests - We got a result of 0.487, meaning the internal consistency was unacceptable. Plot Lord quests - We got a result of 0.638, meaning the internal consistency was questionable. With these results, we decided to not make the average of the results and instead analyze and compare each question individually.

To do this analysis, we first needed to find out why kind of data we had, non-parametric or parametric. So we conducted Tests of Normality, namely the Shapiro-Wilks test. Having conducted this test we had all levels of significance (the p-value) lower than 0.05, so we conclude that the data does not fit the normal distribution, and therefore it’s non-parametric.

According to [2], the test that should be done when analyzing repeated measures and non-normal data is the Wilcoxon’s rank paired test. From the 6 questions made, only the Quest Cohesion question had a significance value less than 0.050, meaning it was the only one where there was a statistical significance between the two sets of data.

**Quest Enjoyment** A Cronbach Alpha test was conducted to measure the internal consistency of the questions. It was done for both the questions related
to the Bannerlord quests and to the Plot Lord quests. They got a result of 0.840 and 0.832, respectively, meaning the internal consistency was good. With these results, we decided to make the mean of the results and analyze those means. Similarly to the quest quality, the test we needed to do for this part was a Wilcoxon’s rank paired test.

The Wilcoxon rank paired test gave a significance of 0.011, which means we should reject the null hypothesis, this is, there is a statistical significance between the two sets of data. We got an effect size of -0.457, meaning the effect...
is average. From the box plot, we see that the players enjoyed the Bannerlord quests slightly more than the Plot lords quests.

Discussion Discussing now the results, in general, we got what we expected since the testing conditions weren’t perfect. The ideal tests would have been done with players already experienced with Bannerlord and capable of testing more than two quests from Plot Lords so that they could experience the variety of quests the mod is capable of providing. With only two quests, this was an impossible metric to measure. It’s possible the test results would change given more time and experience with the game and the mod. It’s also important to notice that everything from the mod was automatically generated, and all the NPC dialogues and step descriptions were written by us, so some English from the descriptions and such could not be perfect, something that could affect user experience.

Regarding the Quest Length, most players though the quests had the right size and there was no statistical difference between the two sets of data, there were, however, a few that though the Plot lords quests were a bit on the small side. This is a great result, since the chosen quests for Plot Lords were purposely chosen because they were on the smaller side. Looking at Quest Cohesion, there was a statistical difference, meaning the players thought the Bannerlord quests were a bit more structurally sound than the Plot Lords quests. Taking a look at the Quest step Clarity. There was no statistical difference between the two sets of data, so both quest pairs had good step clarity, this is, each step in the quests was easy for the player to understand, and they knew what they needed to do. Next, Quest Description Clarity. Again, there is no statistical difference between the two sets of data, so both quest pairs had good description clarity, this is, the dialogue and descriptions provided by the quest giver to the player were good and understandable. Afterwards, the Quest Description Immersion. Once again, there is no statistical difference between the two sets of data, so both quest pairs
had good dialogue immersion, this is, the dialogues proved to be immersive to
the players. Finally, the opinion on the quest origin, this is, if they were made
by humans. Once again, there is no statistical difference between the two sets
of data, so the players didn’t notice any difference between the quests, and they
thought both of them were made by humans.

Concerning the quest enjoyment. We didn’t get the expected results, but they
were still positive considering the challenge we had to face, since the quests were
automatically generated and from what we saw above there are a few problems
with the quest cohesion that might have affected the user experience. There was
a statistical significance between the two sets of data, so the players enjoyed the
Bannerlord quests slightly more than the Plot lords quests.

7 Conclusion and Future Work

The main goal of this thesis was to implement a generator capable of creating
quests that could match human-made quests and even be mistaken with them,
with more diversity and a much more varied structure. To achieve this we ad-
justed the generator envisioned by Jonathan Doran and Ian Parberry with the
modifications done by António Machado to fit a game with a quest system al-
ready in place, Mount & Blade II: Bannerlord. These adjustments consist of: an
adaptation of the number of actions to fit Bannerlord, a change to the algorithm
that assigns the world objects to the actions, the addition of a behavior tree
system and the addition of alternative ways to complete the quest.

The results we got were overall quite positive. We were able to produce quests
that, even though were slightly inferior to the Bannerlord quests in terms of
enjoyment, were comparable in terms of quality, only noticeably less good when
it came to quest cohesion. The automatically generated quests proved to have
a good overall quality and were enjoyed by the players. They also proved to be
indistinguishable from human-made quests which is our most positive result and,
as said before, demonstrates the system could be applied to another game and
the players could not even notice the quests were being procedurally generated.

With these results, we believe the Bannerlord Quest Generator was able to
create quests with great diversity and varied structure that players enjoyed. We
think this is a step forward in the gaming industry, since the generated quests
helped in improving the replayability and enjoyment of Bannerlord, and much
like Bannerlord, this Generator could be applied to other games (with or without
a questing system) to help improve their replayability and reducing the time and
financial costs associated with quest creation.

The best way to continue the work we’ve done is to finish implementing all the
actions proposed by Jonathan Doran and Ian Parberry with the modifications
done by António Machado. We believe the best environment to accomplish this
would be in a game that was done from the ground up with a procedural quest
generation system.

The storytelling part of the quests is also not perfect. So further work on this
component would also allow for even greater quests to be generated.
References

2. Flowchart to choose a statistical test, https://www.intro2r.info/unit3/which-test.html, last accessed 26 October 2021

Game References