Norwegian University of
Science and Technology

noMemory

Author(s)

Bjørn Nødland Fuglestad
Bendik Kiste Hillestad
Per-Arne Waaler Stenshagen

Bachelor in Game Programmering
20 ECTS
Department of Computer Science and Media Technology
Norwegian University of Science and Technology,

18.05.2016

Supervisor(s) Mariusz Nowostawski

Sammendrag av Bacheloroppgaven

Tittel:	**noMemory**
Dato:	18.05.2016
Deltakere:	Bjørn Nødland Fuglestad Bendik Kiste Hillestad Per-Arne Waaler Stenshagen
Veiledere:	Mariusz Nowostawski
Oppdragsgiver:	Norwegian University of Science and Technology
Kontaktperson:	Bjørn Nødland Fuglestad, bjorn.n.fuglestad@gmail.com
Nøkkelord:	Spill, Programmering, Strategi, Tur-Basert, Latex, C++, Blueprint, Unreal Engine, aIMT
Antall sider:	359
Antall vedlegg:	18
Tilgjengelighet:	Åpen

Sammendrag:	noMemory er et prosjekt for å lage et strategi spill i Unreal Engine 4. I dette spillet kontrollerer du én eller flere helter og enheter mot en annen spiller lokalt, eller mot kunstig intelligens. Denne avhandlingen vil gå gjennom dette spillet fra sin fødsel som en idé, gjennom dens implementasjon, og konkludere i våre tanker om resultatet og ferden der.

Summary of Graduate Project

Title:	**noMemory**
Date:	18.05.2016
Authors:	Bjørn Nødland Fuglestad Bendik Kiste Hillestad Per-Arne Waaler Stenshagen
Supervisor:	Mariusz Nowostawski
Employer:	Norwegian University of Science and Technology
Contact Person:	Bjørn Nødland Fuglestad, bjorn.n.fuglestad@gmail.com
Keywords:	Game, Programming, Strategy, Turn-Based, Latex, C++, Blueprint, Unreal Engine, aIMT
Pages:	359
Attachments:	18
Availability:	Open

Abstract: noMemory is a project to create a strategy game in Unreal Engine 4. In this game, you control one or more heroes and units, and battle against another human player locally, or against an Artificial Intelligence. This thesis will go through this game from its inception as an idea, through its implementation, and conclude with our thoughts on the result and the journey there.

Preface

We would like to thank Mariusz Nowostawski for being our supervisor throughout this project, and for aiding us with the game design, report structure and writing. Thanks to Catharina Bohler from Sarepta Studio AS for aiding us with game design and testing, and to Amund Lieng for helping us with testing and balancing. Thanks to Epic Games Inc. for providing the Unreal Engine, and the Unreal community for providing tutorials used during the project. We would specifically like to thank Unreal Engine wiki user Rama for his Victory plugin code, and Tom Looman for his multicolor outline material.

Contents

List of Figures

List of Tables

1 Introduction

Introduction to project NoMemory, or NOMEM. Terminology used in this thesis can be found in Appendix A.

1.1 Project Description

1.1.1 Background

Prior to this bachelor thesis, the authors had collaborated on four other projects in the Game Programming course, and felt that it would be natural to continue working together.

The aim was to find and explore combinations of game genres and mechanics that the authors had not previously seen together, in order to determine if this could result in a game with interesting and fun game-play. The combination decided on, therefore, was an amalgamation of various game-play that each group member enjoyed - focusing on that which in theory could synergise well - and thus providing a unique and fulfilling experience. This design also ensured that each group member had something they wanted to pursue.

In short, the high-level view of the design was largely divided into two major pieces; a strategic "over-world" mode which draws from the two "Lord of the Rings" games "Battle for Middle Earth" 1 and 2, with focus on the "War of the Ring" game-mode in the second game, as well as an action-adventure "dungeon" mode where the player controls a hero who is accompanied by a small group of soldiers carried from the over-world.

1.1.2 Motivation

We have been passionate gamers since childhood, and over the years have also come to embrace the art of programming, so it made sense to combine the two by enrolling in the Game Programming course at NTNU Gjøvik. Through these three years, we have been taught what we need to create a functioning game. We are, however, lacking experience with putting that knowledge into practice and make a proper game, something we endeavoured to achieve in this project.

During the early planning for this thesis, the evaluation of creating our own engine versus using an existing one reached the consensus that using an existing engine would allow more focus on game-play and mechanics. This would make it easier to create a playable game rather than a tech-demo, which is what our previous projects had all ended up as. However, which engine to choose was still an open question. Ultimately, it landed on using Unreal Engine 4, for reasons that are further detailed under 3.2.4.

Another reason for using an existing engine is, simply put, learning. With the sheer complexity of modern game engines, creating one yourself is proving to be a less attrac-

tive endeavour, while learning how to use and adapt to new engines is highly rewarding. Because none of the authors had any prior experience with this engine, it forced us all to learn and adapt quickly, providing a valuable experience for all of us.

1.1.3 Goals

Result goals

- Produce a functional prototype of a game demonstrating unique mechanics and/or storytelling.
- Good communication between group members and well documented code, providing understanding of everyone's code at any given time.
- A clear cut and pursued coding- and documentation standard within the group.
- Get a grade of B or better.
- Minimise our reliance on the supervisor's guidance, and instead independently evaluate its relevance and applicability on a case-by-case basis.
- Write a well-structured thesis with cogent argumentation, appropriate use of lexicon and meta-literary answers.

Effect goals

- Acquire a deeper understanding of the Unreal Engine.
- Improve our understanding of game design and game testing.
- Deepen our knowledge of the C++ programming language.
- Familiarize ourselves with the entire process of developing a game.

1.2 Project Audience

This thesis is written primarily for students starting work on their game programming bachelor thesis, or, to rephrase this, the authors' selves five months ago. It is designed and written in such a way that students wondering more about initial game design for strategy games or technical design for Unreal Engine 4 may read this thesis and learn from our mistakes and successes during the process, as well as get an overview over how much may change from the initial plan to the final product.

The game itself is designed entirely for us as a learning experience, and will not be sold or otherwise mass distributed. It will be available purely as an educational resource.

1.3 Project Organisation

1.3.1 Academic Background

The authors have all attended the Game Programming course, and are all thereby experienced in programming in C++, with Bendik holding the deepest knowledge of the language amongst the three. Through the course we also attended the Rapid Prototyping class, designed specifically around the concept of game jams, where Bjørn and Per-Arne used Unity 4.0, thus gaining some knowledge in using and utilizing existing game engines.

In addition to this, Bjørn and Per-Arne have completed the courses IT1 and IT2 in high school where they learned basic HTML, CSS and C# (that are relevant to this thesis), while Bendik has completed Technology- and Science-studies (Teknologi og Forskn-

ingslære) where he was taught some background knowledge regarding how computers operate.

1.3.2 Responsibilities and Roles

For the project a flat structure was assumed due to the size of the group, and the roles were distributed using a method introduced to us by Simon McCallum. Each of the authors will be given the role of one or more animals, each having its own responsibilities.

The roles, as detailed in Q, are distributed as follows:

- Bjørn
  - Bear
    - Responsible for contacting the supervisor.
  - Owl
    - Responsible for meeting notes.
- Bendik
  - Dog
    - Responsible for making sure all ideas are thoroughly examined.
  - Wolf
    - Responsible for forcing everyone to contribute ideas.
- Per Arne
  - Cat
    - Responsible for removing bad ideas.
  - Rabbit
    - Responsible for room reservations and potentially arranging transport (if needed).

1.3.3 Practises and Rules

The authors have signed a project agreement prior to starting the project start. The rules that the authors have mutually agreed upon are:

- There is no designated group leader, everyone has an equal say in all matters.
- Signing of documents on behalf of the group should be done with the presence and consent of all members of the group.
- Sanctions against a member who has not fulfilled their quota on work done, will be discussed within the group and adjusted according to the scale of the failed quota.
- For a group member who does not meet expectations continuously, expulsion from the group will be considered. For expulsion to happen all the other members have to agree, as well as the supervisor.
- If a member is unable to attend a meeting, this should be communicated in advance.
- If a member of the group disagrees with a decision, the decision will be discussed and voted on. As we are an uneven number of people, matters where only one of

two decisions can be made will always be concluded.
- If a purchase is to be made, it needs to be discussed with the other group members for everyone to split the bill. If the person purchases without asking the group, he will have to pay for the expenses himself.

In addition to these rules, group members were also expected to use time logging programs of their choice to keep track of time spent working on the project. The programs used are discussed in Development Process 3.1.

Any code written would follow a pre-determined coding style, as discussed in Development Process 3.5.3.

1.4 Development Plan

1.4.1 Software Development Methodology

We decided on the agile development model Scrum. This is because it is a development model that we know very well. The weekly sprint meetings also ensure that progress is made, which is essential in the project. As it is an agile development method it is also more capable of handling changes to the project.

The way we will be implementing Scrum is that we will have sprints going from Wednesday to Wednesday. On Wednesdays we will have the sprint review in person for discussing the sprint and planning the next one. For further discussion on Scrum see section 3.3.1.

1.4.2 Schedule

At the beginning of the project, we made a Gantt chart built on the Scrum method, based on what we had hoped we would implement at the speed we had planned. However, this was made without any prior knowledge with Unreal Engine, and the plans were altered heavily during development. A second Gantt chart was therefore made to reflect the true scope of the changes we made. Both charts can be found in Appendix F.

1.5 Document Structure

The document is structured into 10 chapters with appendices at the end:

1. Introduction Introduction to this paper and its content.
2. Requirements What requirements did the game need to reach.
3. Game Design The initial plans for this game, as well as what changed underway during development of the game.
4. Technical Design Describes the technology behind the game.
5. Implementation Explains how game mechanics and features were implemented.
6. Development Process Details what tools where used, how they were used and how communication and integration was handled during development.
7. Deployment How the game was packaged and published.
8. Testing Tests we performed on the game.
9. Discussion The authors own thoughts about the process of the thesis and the game itself.
10. Conclusion Final assessment of the thesis.

2 Requirements

Chapter containing information about the requirements the game should meet.

2.1 Functionality

The following functionality is required from the game:

- At least one playable map
- Minimum left-click to attack game-play
- Single-player functionality
- Aesthetics:
    - Some trees
    - Visible grass
    - Different terrain sections
    - Photo-realistic lighting and shadows
    - Dynamic time-of-day
- AI capable of attacking and moving
- Testing done by at least one classmate and at least one external contact

More on these topics are explained in the 4.1.1 Initial Design section.

2.2 Usability

The game would need to contain some form of tutorial, so that players who are new to the genre may learn and adapt quicker. In addition, a tutorial video should be found under the play menu. The key bindings should also show the user each key's functionality. The game should be familiar with those that have played similar genres before.

2.3 Reliability

The user should be able to play through any game mode without the game crashing. It is expected that the program should be able to run for more than 5 hours without breaking.

2.4 Performance

It is expected that the game will run around or above 60 fps on a modern gaming PC or laptop. On lower end machines it is expected that the performance will be a lot worse as just the engine itself is demanding. If possible, different graphical options should be implemented to help alleviate some of the problems with performance on lower-end hardware.

Based on the specifications from Epic Games' hardware requirements [1], we recommend as a minimum the following hardware:

- **Processor:** Quad-core Intel or AMD, 2.5 GHz or faster
- **Memory:** 8 GB RAM

- **Graphics card:** DirectX 11 compatible graphics card

And the following software:

- **OS:** Windows 7/8/10 64-bit
- **DirectX:** DirectX End-User Runtimes (June 2010)

2.5 Licensing, Laws and Regulations

The authors must follow the licence for any resources used that are not their own. This includes all APIs, third-party libraries, code libraries, textures and other assets that are not already a part of the engine or the starter content. If there are resources without licences, credit must be given to the creator. Depending on the resource, credit is given in the code, as well as in this thesis. Code from forums or web pages is not required to be credited if they have been rewritten during development. For all tools that are used, the accompanying laws and licences are followed.

3 Development Process

This chapter is the summary of how work was done, the tools that where used and how communication between members worked.

3.1 Working Hours

Because of the high requirements for the Unreal Engine, it became unfeasible to meet and work with laptops. Because of this, it was decided that work would be done from home instead, where capable desktops were available. The working schedule was decided independently, matching the required work to resolve the issues for that week's sprint. It was also required to track the time spent working during the project, and the actual time spent can be found in Appendix J.

3.2 Development Tools

3.2.1 Version Control

The authors use Git as the revision control system, as this was preferred by all members of the group. Another option was Mercurial, but none of our members have experience using it. Therefore, Git became the logical choice.

For online hosting of the Git repository the two main platforms that were considered were Bitbucket and GitHub. This was so that the repository could be synchronised between all group members, allowing greater cooperation and productivity. Both were viable options, with not too many differences in terms of functionality. GitHub can be connected with Trello and Slack for eight dollars a month, while Bitbucket can be connected with JIRA and Confluence for free through the school. Including this, Bitbucket also offers unlimited private repositories, while GitHub is capped. In the end, it was decided to use Bitbucket as the host site for the repository.

Because Bitbucket has a hard cap of 2 GB on the repository size, we quickly ran into some issues due to the large size of the third party assets that the project relied on. To get around this, large static third party assets were removed from the repository and added to the git ignore file. The assets were then shared using Google Drive, which each author had to download to their local repository.

Branching

There were two main branches in project repository; The master branch and the development branch. The master branch would only hold completely stable versions of the game, and was used as a point to create testing executables from. The development branch would be the branch the authors create implementation branches from. For each major feature, improvement or content change, an implementation branch was created from the development branch in order to keep the development branch somewhat stable and consistent. The issue was then worked on in that branch and merged into the development branch when completed. When to merge from an implementation branch to

the development branch was up to the individual team member. Each new feature added to the development branch would be tested by another member for bugs or errors, before being completely accepted. When the development branch was stable with major changes to features and content, it was pulled into the master branch.

3.2.2 Issue Tracking Tool

As the authors have used agile development in the form of scrum, it was useful to have a project management tool that supported that development method. For this it was possible choose between a variety of platforms. The main options were Trello and JIRA. The decision was to go with JIRA, as it integrates with Bitbucket and offers some more complex functionality, though this makes it slightly harder to use. It was mainly used for its scrum board, organizing sprints and issue tracking.

3.2.3 Documentation Tool

For a shared writing document Google Disk folders were used. It was used for creating quick drafts and notes early in development, as well as writing the game design document, project plan and sprint logs. For writing the bachelor thesis itself it was decided to use LaTeX because of its academic standard. Initially, Texniccenter was used, as ShareLatex was not available for free at the start of the project. When it became available after about a month, the choice was made to switch to ShareLatex instead, with the main reason for using it being the cloud integration, allowing for multiple users to edit at once without needing to synchronize through a repository.

Confluence was also decided upon for documenting meeting notes and major decisions such as documentation protocols, as Confluence is in the same package as JIRA and Bitbucket, and is therefore easily integrated.

3.2.4 Engine

Using an engine is recommended when making a game, and so a decision had to be made between writing a new engine or using an existing engine such as the Unreal Engine, Unity, etc. The decision was made against writing a new engine, as that would take too much time away from developing the game itself. This is because the engine would have to be written mostly from scratch. This was a lesson the authors had learned in the Game Programming class the semester before, where they had written more than a hundred source files, and yet could only show off a tech demo rather than a game.

A decision then had to be made regarding which engine to use. There are multiple engines available, but the focus was mainly on Unreal Engine 4 and Unity 5, as they are both built around a business model allowing those who do not want to sell their game to use the engines for free. Both have assets available in their online store and are highly functional engines with many good qualities, such as Unity's user friendliness and Unreal Engine's graphics. Although Unity is considered more beginner friendly, in the end it was decided to use the Unreal Engine, as it uses C++ as the coding language and due to the engine being more advanced, allowing for further in-depth options. The development was done using the 4.10 version of the engine, as this was the newest version available at the beginning of the project.

3.2.5 Compiler

Visual Studio 2015 has been used in tandem with the Unreal Engine to manage any C++ code that was written for the project. The main reason for this is because it is used by default by Unreal Engine, and it is the compiler that the authors have used the most during the Game Programming course.

3.2.6 Communication Tools

Skype has been used as the main means of communication between the authors, as Skype has been used by all for years. It was also used for group meetings through Skype call when the authors could not gather at one place due to sickness or other. As they are such a small group there were no problems with using Skype, despite it being a private form of communication. E-mail was also used for communicating with the supervisor and the external, due to the more sporadic communications with them.

3.2.7 Misc Tools

- Microsoft Project was used to create Gantt charts.
- Microsoft Paint was used to create initial map drawings.
- GIMP 2 and Paint.net was used for cropping, scaling and arranging images before inserting into the LaTeX document.
- Toggle and Excel for time logging

3.3 Project Workflow

3.3.1 Scrum

Scrum is an iterative and flexible software development framework with focus on frequent meetings and collaboration. A cornerstone of Scrum is its recognition of and ability to deal with the volatility of the project requirements. This allows Scrum to maximize the ability of the team to deliver quickly, which was important for the project.

See bibliography entry [2] for more details on Scrum.

Our implementation of Scrum had minor alterations, however, as there were no standard scrum masters. The sprint review, reflection and start meeting were all merged into one meeting, as dividing them up would be a waste of time with such a small team.

3.4 Development Workflow

All meetings were held on Wednesdays, as this was a day where everyone was fully available. Because of this, the meetings were held back to back, combining sprint start and end meetings. This also produced a combined Confluence log. The meeting summary was sent to the external and to the supervisor if he was not present.

3.4.1 Sprint Start

Each sprint started with a sprint meeting. Every member chose issues from the backlog and moved them into the sprint, or created new issues directly into the sprint. Story points had to be added to all issues, to reflect the amount of work that was expected to be done to complete the issue. The issues were also added to the sprint log as goals, as well as how we planned to resolve them. When all members had been assigned issues and were happy with their workload, the sprint would start. After this point, no more

issues could be added to the sprint. Although it is possible to add issues to a sprint in progress in JIRA, it was not done as it changes the sprint scope.

3.4.2 Sprint progress

During the sprint, a member would move his issue from the "To do" area of the scrum board to the "In progress" area of the scrum board when work on the issue was started. When completed, the member changed the issue he was working on from "In progress" to "Resolved" or "Completed" on the scrum board. Most issues would go to the "Resolved" area, as issues placed there may be re-opened, should the group decide in the future that the issue was not completed in a satisfactory way.

3.4.3 Sprint end

At the end of the sprint, a review meeting was held. During this meeting, the sprint log would be updated with the results of the sprint and the authors' reflections. If there were any unfinished issues at the end of the sprint, it would be added back to the backlog and picked up again later.

3.4.4 Help

If a member was stuck on a particular problem, he could ask the other members through Skype. If the problem was not solvable using Skype, all authors would look at the problem again during the sprint meeting. If this would not work either, the authors would discuss the issue with their supervisor at first opportunity given.

3.5 Organisation Of Quality Assurance

3.5.1 Testing

Testing of the game is needed to ensure an enjoyable player experience. The main features that need to be tested for the game to actually work are performance and reliability, while other tests are needed to ensure better quality of balance, mechanics and aesthetics, which requires play-testers. More on testing can be found in the Testing 8 chapter.

3.5.2 Documenting

The code has been documented in such a way that it is possible for other team members to understand and start working on a part of the game that they have not worked on before. All of the code written in C++ features Doxygen-style comments, so that it could automatically be turned into a reference manual and added to the appendix at the end of the project. The manual can be found in Appendix R.

Blueprints have been documented by adding comment blocks around segments in the Blueprint, describing their function and relation to each other.

3.5.3 Coding Style

When developing it is useful to use a common programming style to make the code more reliable and easier to maintain, read and debug. While it is possible to create our own programming style, the amount of content in the C++ language to cover is very large, and it was therefore easier to use an existing standard, and, if need be, change the parts where the authors disagree with the standard in question. It was originally decided on using the Google C++ programming style, as it extensively covers coding standards in

the C++ language. However, when working with Unreal Engine, it was discovered that they have their own coding style, and though not as detailed as the Google C++ style, it still covered enough areas that it was used during development. Also, when working with Unreal, it is reasonable to follow the style it is written in, which can be found here: [3].

4 Game Design

This chapter contains information about the design of the game. Some further details can be found in the "Game Design" document in Appendix D and in the "Decisions and Notes" document found in Appendix E.

4.1 Game Design Document

The game design document was created during the first few weeks of the project together with the project plan. The purpose of the game design document was to lay down the features of the game that the authors all agreed on. By creating the game design document, the authors would all have an idea of what the game would be like and what to strive towards. By referring to the game design document they could also see which features needed to be implemented, as well as which features were underdeveloped and needed improvement.

The game design document starts by laying out the more broader features of the game before going into more detail surrounding each game mechanic. This is to give the reader an overview of all features, preparing them to delve deeper into the details and specifics of each feature. This also makes it easier to change the document during the development process, as this setup simplifies access to details, and single features may be changed without needing to alter the main scope or other features.

During development there were changes that were needed to be done. Some of the reasons for this were improved design, time constraints or technical limitations. Any iteration to a mechanic was added to the design document underneath the old one. This was done for maintaining the original design rather than replacing it with the new one.

4.1.1 Initial Design

Much of the initial design thoughts revolved around either a full game or a mostly full game, and there were therefore some uncertainty as to what we would be able to implement and what would have to be cut. Despite this, most of what is written here would be implemented in case of a full game.

Game modes

Within the game there were several game modes that had plans for implementation. The main idea was to implement game modes in the same fashion as Lord of the Rings: Battle of Middle Earth 2; War of the Rings game mode. That meant that there were going to be three game modes: A campaign mode, a skirmish mode and a conquest game mode.

The skirmish game mode features a single instance of combat. The player enters combat on one side with his units and the goal is to defeat the other teams units before losing all your own.

The campaign game mode features a world with a story attached to it. While in the world map the player would be able to build more units and choose which regions to

invade depending on the story. The player would follow the story and engage in many battles where the outcome would affect the world map, so that when units were lost in battle they are removed from the world map as well. The objective would depend on the campaign. It could be to capture the enemy capital or capture all their territory.

The conquest game mode would be similar to the campaign except there would be no story or other scripted events. The goal in this case would also be set by the player, rather than a preset story-driven goal. These player-set goals could include total dominion where the player needs to conquer all regions on the map, capital defence where the player needs to conquer all enemy capitals while defending their own capital, or a "survival" mode, where there is one player with an easily defensible position trying to survive for a preset amount of time against multiple foes.

World Maps

The world maps are used within the campaign and conquest game modes. These maps offer a variety of terrain and regions, making most combat events unique. Each region also has one or more specific resources, as well as different starting development and, therefore, available construction plots.

Sizes of different maps vary, depending on amount and size of regions. Larger sizes of regions leads to the ability to support more factions on each map, and allows for more strategic variety.

The campaign is a story driven event based experience, and is designed around the concept of inbuilt tutorial, more challenging tasks as the game progresses, and has certain regions, buildings and units locked, until the player progresses far enough into the story to be considered experienced enough to wield these.

Conquest uses the same maps as the campaign and more, but has all regions, buildings and units unlocked from the very beginning. This is designed for those who have completed the campaign, and are looking for more challenges.

Figure 1: Example of campaign map

When two or more opposing armies meet in the same region, with at least one of the armies in question being controlled by a human player, the game zooms in to that region and starts the combat phase.

Combat

In this phase, the players take turns to move their units. During their turn, the player may move units and heroes a set distance, as well as attack enemy units, cast spells, and use various items. These spells and items were undecided, and would only have been implemented into the game if the time constraints allowed it and game testing demanded it. These spells and items would have been basic, however, and would most likely have included potions, attack boosts, defense boosts or stamina boosts.

It was initially uncertain whether units should move on a grid or freely around the map, and also whether to use action points or simply a set number of movement points. If the latter was used, units would be limited to only one attack per turn, but with the former, units may attack as long as they have the action points necessary to do it. In the end, it was decided to use free movement together with action points.

Each unit should have a clearly defined health-bar, to avoid any guess work by the player into the amount of health a unit has left. Specific forms of attack and defense, such as crush, pierce and shock could also be implemented.

Units

The actual units, their statistics and aesthetics were mostly outside of the authors' control, due to the reliance on the assets available in the Epic Games marketplace, as well as other similar stores or forums. However, as many unit types as possible were intended to be implemented. These units include:

- Swordsman
  - A basic sword-wielding melee unit.
  - May be equipped with a shield for extra defense.
- Archer
  - A basic ranged unit.
  - May be equipped with different types of bows for a wider variety of range and attack strength.
- Spearman/Lancer
  - Usually a melee unit, but may throw spear/lance at range. Penalty for this action is undecided.
  - Extra damage towards mounted units.
- Rider
  - Mounted unit.
  - Spends fewer action points on movement
  - May dismount.
- Rogue/Assassin
  - Stealth unit.
  - May equip a personal "fog of war", that requires enemy units to get in close to see the unit.
  - Has a ranged and a melee attack, but neither is as strong as those of swordsmen or archers.

- Monster/Berserker/Tank
    - High damage and high health, but slower movement/ fewer action points/ fewer movement points.
    - Area of attack damage, hits all units within a certain radius.
- Mage/Caster
    - Magic user, may heal self (and possibly others).
    - Extra damage against heavily armoured units such as monster/berserker/tank units.

Units may also be designed to work in groups or cohorts, giving them more attack strength and health, as well as group/cohort specific abilities, such as shield wall (for added defense), wedge formation (for added attack power, but reduced defense) or spear wall (for added strength, but removed movement).

Heroes

Hero units would have many similar abilities and aesthetics as their regular unit counterparts, but with greatly added strength, speed, defence and action points/movement points. These units would also be given a specific aesthetic detail singling them out from a crowd, such as a halo, a crown or simply making them look bigger.

Various suggestions for types of hero units are the same as for regular units. This includes groups/legions, by turning them into heroic armies/legions. It has also been suggested that if the hero or heroes die in a battle, all regular units will automatically flee, and the battle will automatically be forfeited.

Buildings

The number of buildings available for each region depends on the regions development. Each region starts with a preset development level, with the possibility for the player to spend resources to increase this.

The game is designed around four main buildings:

- Resource generating building (farm, mine, market, tax collector, or other).
    - Generates a resource, for example money.
    - May be expanded to include several types of resources, such as wood, stone, iron and gold, including specific buildings that produces them.
- Unit generating building (barracks, academy, or other)
    - Creates all forms of non-heroic units, such as swordsmen or archers.
    - May be expanded to include several types of unit creating buildings, each creating its own class, such as barracks for melee foot soldiers and archery range for ranged units.
- Stronghold building (fortress, watchtower, guard-tower, or other)
    - Defending units spawn at their stronghold.
    - Ranged units placed on top of stronghold have increased range.
- Hero building (palace, hall of heroes, knights joust, or other)
    - Only building capable of producing one or more hero units.

- Can only be placed in the capital province. If the capital is captured, the building is moved to a new region.

4.1.2 Final Design

This sub-section aims to describe changes that happened during development and why they happened. If a feature is not mentioned in this sub-section, it can be assumed that it did not change from the Initial Design.

Campaign game mode and Buildings

Campaign army movement has not been implemented together with regions. As there are no armies moving we had to get around this by using a set of pre-made scenarios that are played through. Such that when one scenario is finished the player transitions to the next until all scenarios in the campaign are done.

Buildings have not been implemented, and would have been one of the subjects of focus for future implementation, as described in section 9.2. Some empty buildings have been added on the pre-made map, but these buildings have neither textures nor functionality, and therefore only show where and how they could be placed on the map. These buildings are therefore merely used as decorations, although in theory they could also be used for strategic positioning of ranged units.

Combat

When researching how to implement a tile based movement system it turned out to be more difficult than originally anticipated. There was an option to use a third party system for grid navigation, but the authors decided against it, as a grid free movement system was easy to make using the native functionality in Unreal Engine 4. For this reason, a grid free movement system was implemented instead.

When moving a unit it was decided against using movement points and only one attack per turn, as this may have ended up in a flurry of hit and run tactics from the players. Each unit has instead their own pool of action points that they may use, where attacking, moving and all other character actions cost action points. The authors felt that this design simplifies both the implementation of the system itself and the range of interactions available for the user, as a system with multiple value pools for different actions may have made implementation and user interaction problematic.

Unit health may be seen by hovering the mouse over a unit, or selecting it if it is on your team.

Six different damage types have been implemented in the game. These are fire, water, air, earth, blunt and slash. These damages are reduced differently based on the stats of the unit taking the damage.

No special abilities, such as spells, portals or unit stances, have been added.

Units

Certain units could not be added simply because the needed assets could not be found for free, such as bows for archers,spears for spearmen, horses for cavalry and dragons, while seven forms of monsters were included because assets where available, such as trolls, grunts, grunt masters, spiders, half giants and dark knights. Heroes were not implemented.

Grouped unit were not added as they proved to be difficult to make. This, as well as other minor changes, are discussed more in section 9.2.

Each unit has their own set of statistics. These are:

- **Strength** - Increases accuracy with strength based weapons
- **Dexterity** - Increases accuracy with dexterity based weapons
- **Intelligence** - Increases accuracy with intelligence based weapons
- **Constitution** - Increases health and max amount of action points
- **Speed** - Increases movement and action point regeneration
- **Perception** - Increases starting action points
- **Health** - Base amount of health
- **Action Points** - Base amount of action points
- **Action Points Regeneration** - How many action points the unit gains at the start of each turn
- **Fire Resistance** - Reduces damage from fire attacks
- **Water Resistance** - Reduces damage from water attacks
- **Earth Resistance** - Reduces damage from earth attacks
- **Air Resistance** - Reduces damage from air attacks
- **Armour** - Reduces damage from none magical attacks
- **Hit Chance** - Base chance to hit with any weapon
- **Dodge Chance** - Chance to dodge attacks
- **Block chance** - Chance to block attacks
- **Critical Hit Chance** - Chance to acquire a critical hit on an opponent
- **Critical Hit Multiplier** - Amount to multiply damage on critical hit
- **Movement** How far the unit can move per action point
- **Sight** - View distance (fog of war not implemented, so this is purely for future expansion)

The statistics are also able to take modifiers, such as increase health by 20 or multiply by * 1.05. Using this we are able to create weapons that improve the wielders statistics. As well as depending on the statistics, it is also scalable. This added implementation also simplifies future expansions such as items and leveling.

5 Technical Design

This section is about the technologies that were used during the development of the game, with high-level description of how some of them were applied.

5.1 Unreal Engine

The game has been developed using Unreal Engine 4.10.X. A feature of the Unreal Engine is its Visual Scripting language called Blueprint. The Blueprint language itself is very easy to learn if you are familiar with object oriented languages. The engine itself is made using C++ and supports both creating games using Blueprint and C++. Blueprints are fast and easy to make and are appropriate for quickly creating prototypes and some permanent features. Once created, the Blueprint script translates for the most part very well to C++ code, so it can be converted over if needed. There are, however, some functions that are hard to find due to having other names in C++.

There are some times that functionality is not available in Blueprints, and those times it is possible to implement the functionality in C++, and then expose it to Blueprints. There are also plug-ins that provide additional functionality that can be added to the engine.

5.1.1 Utilisation

Among the areas where Blueprints were exclusively used, were the camera, the level-streaming and the procedural foliage logic. This was largely due to the relative simplicity of implementing this kind of asynchronous functionality in Blueprints as opposed to C++. There were also no noticeable performance losses from using Blueprints in this case, as most of the actual processing involved was handled by the Unreal Engine's implementation of the functions used.

5.1.2 Level-Streaming

From the Blueprint side, the level-streaming was quite trivial, consisting primarily of an event followed by the invocation of the engine's level-streaming code. The event would be fired by a trigger volume inside the game-world which would respond to the camera entering or leaving the volume. Most of the work, therefore, involved placing appropriate trigger volumes and splitting the game-world into chunks that the engine could load. The Blueprints for the levels themselves, which handled game-play logic for that level, also had to be updated to be aware of this loading/unloading mechanic. This involved sharing a global game-state, which is further detailed under Drawbacks.

5.1.3 Procedural Foliage

For the foliage in the game it was decided to utilise procedural generation, due to the sheer amount of manual labour that would otherwise be required for a large map. This is mostly done using features the Unreal Engine provides, such as the simulation of multiple generations of tree growth. However, Blueprints are required to tie the necessary

components together, and they are also used for describing the landscape material which ultimately decides where grass can grow.

5.1.4 Drawbacks

A drawback from using Blueprints comes from the fact that it is so heavily object-oriented, and so in order to implement a persistent global game-state that can be referred to from different Blueprints, one has to create an Actor subclass that is instantiated somewhere in the game-world. This actor is then referred to via a magic constant in the Blueprints, in order to access the state variables it encapsulates. The actor does not implement any functionality, and so it would make more sense to have implemented this game-state as a struct, or just a group of global variables, but this is however only available in C++ code.

6 Implementation

While the implementation of the game's features was a collective effort, the workload had to be divided into discrete tasks, so as to avoid conflicts when merging. Because of this, several different tutorials were used[4–10], and some ways of implementing the design may differ.

6.1 C++ and Blueprint integration

6.1.1 Expose Code To Blueprints

For Blueprint scripts in this project to be able to interact with the C++ code Unreal Engine's macro features are needed.

```
UENUM(BlueprintType)
enum class EAttribute : uint8

USTRUCT(BlueprintType)
struct NOMEMORYPROJECT_API FUnitStatDataRow

UCLASS()
class NOMEMORYPROJECT_API ABaseCharacter
```

In addition to the small syntax changes it is also very important to include the following macros as shown above and if it is a class it must be a subclass of UOBJECT.

- UCLASS()
- USTRUCT()
- UENUM()

These macros are needed to gain access to structs, classes and enums in Blueprints, though specifiers are needed to suggest to the engine how they may be exposed in the editor. The ones that were used the most were:

- BlueprintType: Exposes this struct/enum/class as a type that can be used for variables in Blueprints. For classes this means that it is possible to set an instance of the class as a default value in the editor. Without it, the default pointer value would be null.
- Blueprintable: Blueprint classes may inherit from this c++ class. Any class that inherits from this class in C++ will also inherit this ability. That is why when inheriting from AActor in C++ this specifier is not needed as it is already included. The specifier was used when inheriting from UOBJECT.

This covers how objects were exposed to Blueprints. However, access to variables, properties and functions from the Blueprints is required as well. To do this, two more macros are added:

- UPROPERTY()

- UFUNCTION()

As with the object specifiers these also need specifiers to tell how they are to be exposed. For UPROPERTY, the following specifiers were used the most:

- EditAnywhere: Indicates that this property can be edited by property windows, on archetypes and instances.
- BlueprintReadWrite: This property can be read or written from a Blueprint.

```
UPROPERTY(EditAnywhere, BlueprintReadWrite,
Category = "Unit|Animation")
    bool bDead;
```

For UFUNCTION, the following specifiers were used the most:

- BlueprintCallable: The function can be executed in a Blueprint.
- BlueprintPure: the function does not affect the owning object in any way and can be executed in a Blueprint. Mostly used for getter functions.

```
UFUNCTION(BlueprintCallable, Category = "Unit")
        void CharacterDeath(float Delay);
```

The specifiers that have been mentioned here are but the tip of the iceberg, but these are the ones that were mainly needed for exposing the code to Blueprints. It is also worth mentioning that private variables and functions cannot be exposed. To get private variables, expose the get functions.

6.1.2 Call Blueprint From Code

A problem arises when it is needed to communicate from a C++ class to Blueprints. Since Blueprints are made on top of C++, there is now way to access Blueprints directly in C++ code. To get around this, a base class for the Blueprint in created in C++ and exposed to the editor. The default values can still be set in the Blueprint, and if a component is added, it can also be edited in the Blueprint.

Sometimes performing a function in Blueprints was desirable over performing it in C++. In this case, functions were used with the BlueprintImplementableEvent specifier. These functions are created in the header, but have no implementation in the code. It is then possible to override them and implement them in Blueprints. This has enabled C++ code some way to call functions in the Blueprints. BlueprintNativeEvent could be used as well, which is similar to BlueprintImplementableEvent, with the exception that it also has a C++ implementation.

6.2 User Interface

All UI in this game was created using **UMG** (Unreal Motion Graphics) which is a visual UI creation tool within the Unreal engine. The core of UMG are Widgets, which are a series of pre-made functions that can be used to construct interface, such as buttons, checkboxes, sliders, progress bars and more. The widgets are editable within a Widget Blueprint. Within each Widget Blueprint there is a design tab for creating the UI layout and a graph tab for creating the widget's functionality in blueprints.

Depending on the widget different events are available to trigger in the widget. When the events are created they are added to the event graph of the widget Blueprint. Then

all that is needed is to implement the functionality of the event. It is also possible to bind functions to widget values as seen in Figure 2. The bind function is on creation added to the widget Blueprint as a function returning the expected value. The value that has the bind function will then updates itself by running the function.

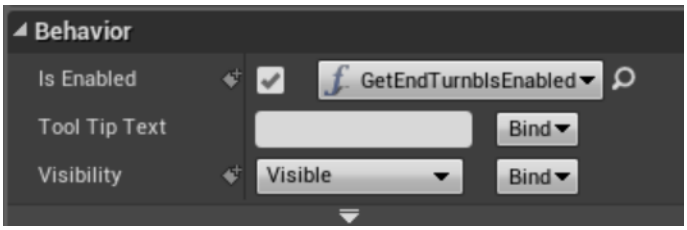


Figure 2: UMG Bind Function

6.2.1 Main Menu

The Main Menu is added to the viewport when the player starts the game and enters the Main Menu level. To navigate the Main Menu, the buttons at the bottom of the screen is used. When pressed, the current menu widget is removed from the viewport and the widget for the new menu is added to the viewport. This design was inspired by the menu from Lord Of The Rings: Battle For Middle Earth see Figure 3.

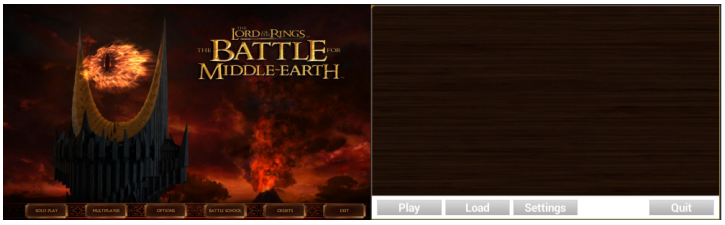


Figure 3: Menu Comparison

In the Settings menu, tabs were used to change between each of the categories. Each category is its own widget Blueprint that was designed and implemented separately before being added to the Settings menu widget. The design of the Settings menu is inspired by Diablo 3 see Figure 4.

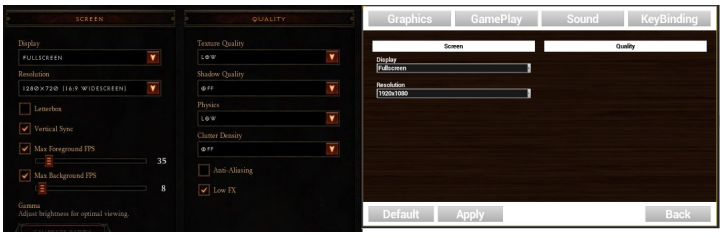


Figure 4: Settings Comparison

To start games from the Main Menu, you navigate into the Play menu. Here you can select a game mode; Campaign or Skirmish. While Conquest is added, it is disabled, as the game mode is not implemented. Both Campaign and Skirmish use the same template

widget, but implement their own buttons for choosing the level to play. When the player presses Start, the game loads the selected level, and the selected difficulty is saved then reloaded when the game starts.

6.2.2 Game Menu

Skirmish

For the skirmish, UI it was divided into several layers using an overlay widget. The reason for doing so was because several of the layers where overlapping, it also makes it easier to disable or hide a the layers separately.

Unit canvas contains information about the selected unit and the unit that we hover over. The values displayed the are updated by the player controller. When the player hovers over a character, that characters name and health is displayed in the health bar on top of the screen. When the player in not hovering over any character the bar is hidden. When the player selects a unit the unit canvas is updated with the statistics of the selected unit. The health and action points are added to the action bar. While the other statistics are added to the side tabs along with the weapon statistics as show in Figure 5. The in depth stats can then be toggled using the side arrows on the action bar.

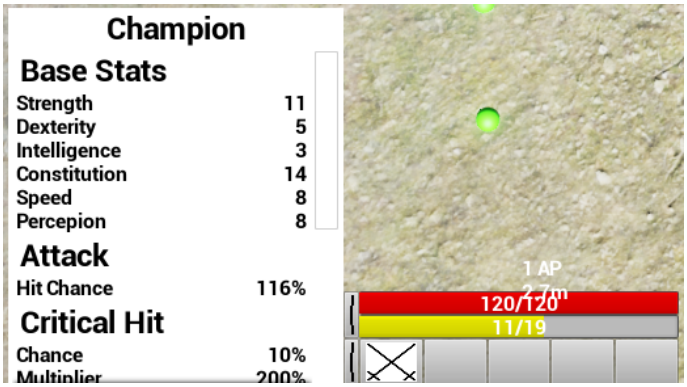


Figure 5: Statistics Menu

Info Canvas displays buttons for menu and next turn as well as the combat log.

Menu Canvas is displayed when the menu button is pressed. The menu canvas is hidden by default. When opened the other layers are disabled and the game is paused. From the menu it is possible to exit to the main menu or exit the game. It is possible to return to the game by pressing the resume button. The resume button hides the menu layer again and re-enables any layers that it disabled, it also un-pauses the game.

Victory canvas is displayed when a lose or win condition is reached. The game mode triggers a function in the UI that ends the game by pausing the game and hiding all the other layers. The victory screen displays weather the player has won or lost. The only option left is then to exit to the main menu.

Combat log

The combat log is a multi line text widget wrapped in a scroll bar. Whenever a line of text is added to the combat log the scroll bar is automatically lowered to the bottom. When the characters in the combat log reach above a thousand the first hundred are removed to from the text line as a slight optimisation. To add new lines to the combat log the characters calls a function in the skirmish UI and pass the string to be added with it.

Floating text

The floating texts are used for displaying the game information in addition to the combat log. The event triggers in the character on events such as take damage. The floating text widget is created and added to the viewport as seen in Figure 6. The damage is then turned into a string an pass and set as the text in the floating text widget. The Floating texts position is converted to the screen location and its animation is played. After a time limit the floating text expires and the widget is removed from the viewport.

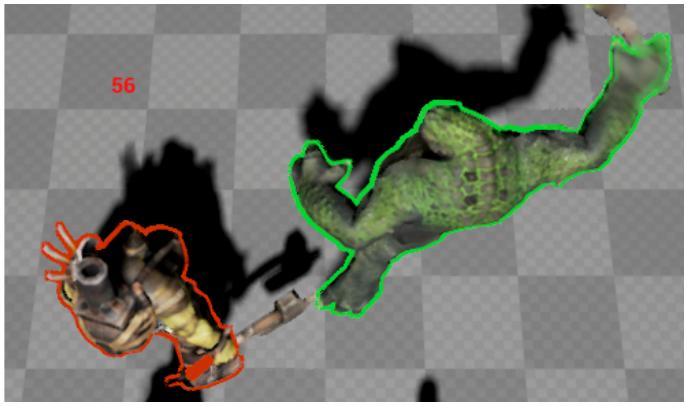


Figure 6: Floating Text

6.3 Controllers

Controllers in Unreal can be thought of as the strings on a puppet or soul in the game, being able to possess different bodies and control them. In this case, the player controller is the way for a player to interact with the game world, and input from the player is taken in by the player controller and executed.

6.3.1 Skirmish Controller

The skirmish controller is able to select characters using the mouse. When the left mouse button is pressed, a line trace is done below the mouse position. The hit result is checked to see if a BP_BaseCharacter was hit. If a character is selected, a reference is stored in the skirmish controller. In the case that no character was hit, and if there is a currently selected character, the selected character is deselected. If another character was hit, the selected character is deselected and the one hit is selected.

The left mouse button performs an action if there is a selected character. Again, a line trace is done to get a hit result. If no character was hit, the hit location is acquired. If the character can move to the location, the location is sent to the character's AI controller, which then performs a simple move to the location and reduces the character's action

points for moving that distance. If a character is hit, an attack action is performed, if possible.

During the tick operation, again, line trace operations are performed under the mouse position. Checks are done if a character or a location is hit. If a character is hit, an OverUnit event is triggered, and if no character is hit, an OverLocation event is triggered. In the two events, the path to the location or the character's location is found. Using the path, a navigation spline is updated to show the path the character will take, as seen in Figure 7. When it is done, the hover over information is also updated. In the case of a location, the distance and action point cost is displayed. In the case of an enemy character, the damage range, hit chance, distance and action points are displayed.

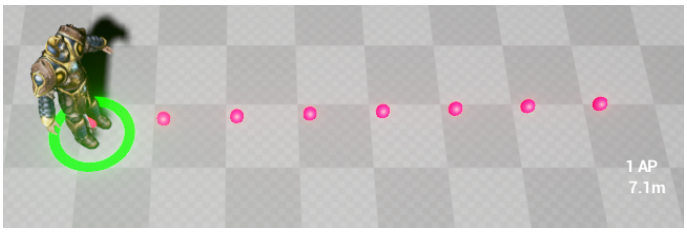


Figure 7: Hover over location

6.4 Game Modes

6.4.1 Skirmish

The skirmish game mode becomes notified when a player ends their turn. It then starts the next player's. When starting the next player's turn, the game mode finds all character belonging to the player's team and invokes the "BeginTurn" event on them. It does this by iteraing over the game world looking for BaseCharacters. To check if the game has ended, in the game mode tick, the game mode checks if the player has lost all of his characters, and if the condition is met, a win or lose event is triggered.

6.5 Characters

Each character in the game is based on the BaseCharacter class. All functionality needed for the characters in the game is implemented in the BaseCharacter. Each character extends the BaseCharacter, and changes the default values. For most of the characters, it was only necessary to change the Skeletal Mesh Component, but for the monsters it was also required to change the collision, as the skeletal mesh might shrink or expand in comparison to the normal humanoid characters. In addition to the components inherited from the ACharacter, two decal components are also added to the BaseCharacter. One for displaying its team color, and one for displaying the range of the character based on the characters equipped weapon as can be seen in Figure 8.

Figure 8: Character showing decals

6.5.1 Statistics system

For each character there is a collection of statistics used for their interactions with each other. Normally, only adding the statistics as values in the character would be enough. However, in the game, the authors wanted to be able to influence the statistics by adding modifiers to the statistics, making statistics linkable and making them scalable. The authors solution for this was to place the statistics into the following inheritance structure seen in Figure 9. For the full version, see Appendix N. The "Get Final Value" is overridden in each child and calculates the final value of the statistic.

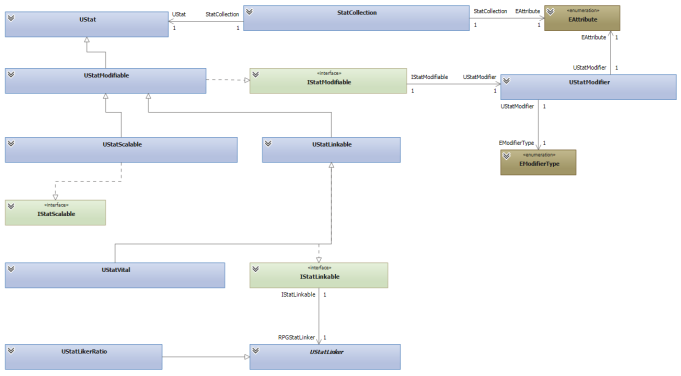


Figure 9: Compressed Statistics structure

Modifiable statistics allow for added modifiers to an array in the statistics. Each modifier has a value and an operation that is used when applying it to the statistics. When calculating the statistics modifier value, each of the modifiers is gone through, and is added to an array where each place stores the value of the same operation. This is because different operations such as add and multiply needs to be applied in different order.

Whenever a new modifier is added to the statistics, the modifier value is recalculated. When the final value is needed, the modifier value is added to the base value.

Linkable statistics are similar to the Modifiable statistics. Each Statistic Linker is added to the array in the Linkable statistic. The Statistic Linkers contain a reference to another statistic that is linked from. Whenever a new linker is added to the statistics, the linker value is recalculated. The final value is calculated from the base, modifier and linker value.

Scalable statistics contain a value that can be set. The scale value is also added to the modifiable calculation. The final value is calculated from the base, modifier and scalable values.

Vital statistics contain a current value that is in the range between zero and the final value.

Initializing the statistics are done using the an initialization function in statistic collection. The function takes a data table as a parameter and the characters unit name as a row name. From the data table a unit data struct is gotten which was used originally when creating the data table as seen in Figure 10. Within the struct are all the default value for that unit. Using the value from the struct all statistics are made and initialized in the initialization function.

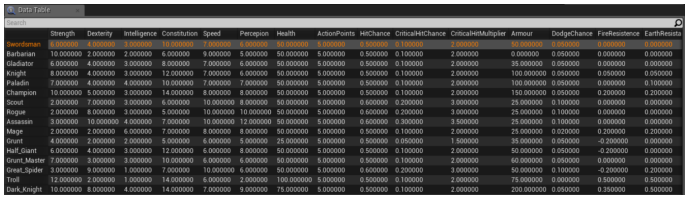


Figure 10: Data table

6.5.2 Equipping

When equipping a weapon, the weapon class is accessed and an instance of that weapon is created. The weapon root is then attached to a socket on the skeletal mesh as seen in Figure 11. When that is done, all the modifiers from the weapons are added to the statistic collection. As there is no system for selecting a weapon to replace it, the main weapon is automatically replaced when both offhand and main hand are full. When a weapon is unequipped, the weapon actor is destroyed and all the modifiers from the statistics are removed.

Figure 11: Weapon Socket

6.5.3 Attacking

When a character is attacking, it triggers the "AttackUnit" function. First, checks are done to see if the character can attack the target:

1. Check if on the same team
2. Check if enough action points
3. Check if within range
4. Check if within line of sight using line trace

Once done, it is certain that the character is able to attack the other character, and another set of steps are then needed to be done. First, rotate the attacker to face the defencing character. Then, the attack animation is triggered by setting the "attack" Boolean to true. See 6.7 for more. After that, the amount of action points are reduced by the amount used to attack with the weapon(s). Then, checking if the target was missed is done by checking the attacking character's hit chance against the other character's dodge chance. The attacking character's hit chance is calculated by taking the weapon's attribute, such as strength, and each point in strength the attacker has increases hit chance by 6%, it is then reduced by the defending character's dodge-chance attribute. A random float between 0 and 1 is created, and a comparison is done to see if the random float is higher than the hit chance. The same is done afterwards with the defender's block chance to check if they blocked the attack. The attacker is then certain to hit the defender. The damage is then gotten from the weapon(s) as well as the damage type. The critical hit is then checked using the critical hit chance the same way as hit chance.

Then, a switch on the main weapon attack type is done. On melee weapons, TakeDamage on the defender is triggered, and the damage and damage type is passed. If the attack is a ranged attack type, the projectile class is accessed from the weapon and spawned in the direction of the defending character. The damage is added to the projectile after it is spawned. When the projectile hits a character, it triggers TakeDamage on that character.

In TakeDamage, the incoming damage is reduced depending on the damage type. The

damage type is cast to the authors' BaseDamage type, and a default object is spawned. "Calculate reduced damage" is then called on the damage type instance, taking damage amount and the units stats. In the case of fire damage, the damage is reduced using the fire resistance stat, and is then returned. The final damage reduces the character's health. If the remaining health is bellow 1, the character then dies.

6.6 Weapons

The base of the weapons in the game is the BaseWeapon class, as seen in Figure 12. Every new weapon then inherits from these classes and add their own skeletal mesh or static mesh component to show the visual weapon. Each weapon has an attack type that describes if the weapon is a melee weapon or a ranged weapon. The weapons also have a damage type class. This means that it is possible to have an axe that deals fire damage. With each weapon there is also an array of modifiers that may be added to the weapon. With the modifiers you can create a sword that gives the player +2 strength, +10% fire resistance or +100 health. All the stat modifiers that the weapon has is added to the character's stat collection when the weapon is equipped, then removed again when unequipped.

Within ranged weapons, a projectile class is instanced when the character attacks with the weapon, and is spawned in the direction the character is facing. When spawned, it ignores the player who shot it, so that they do not accidentally hurt themselves. In the case that the projectile clips a unit while heading towards the target, that character will be hit instead.

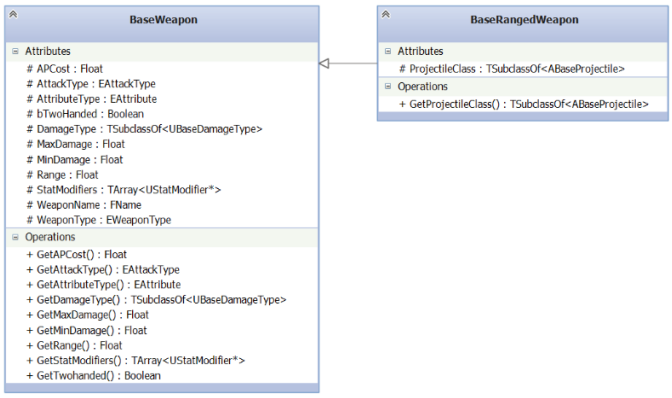


Figure 12: Weapon UML

6.7 Animation

In this project, the authors did not make their own animations, as none of them have any experience as animators. This meant that they relied on the animation assets from the Unreal Marketplace. For managing the animations, Unreal Engine provides Persona[11], which is an animation editing tool-set. In Persona, animations are arranged in the hierarchy seen in Figure 13. At the bottom, there is the skeleton. It contains each bone and joint that can be moved in an animation. After that, there is the skeletal mesh. This becomes the body of the skeleton. Several meshes may share the same skeleton. After that, there are the animations that apply to that skeleton. All skeletal meshes which share that

skeleton may use those animations. Last, there is the animation Blueprint. The animation Blueprint is a specialized Blueprint, as it contains an Animation graph that outputs the final pose for each frame and a normal event graph that drives animations in the animation graph.

Figure 13: Animation Hierarchy

In the beginning, the animations assets that were used came with the characters in the Infinity Blade Warriors and Adversaries, as well as the animation starter pack. For most of the monsters, this was fine, since all the animations that were needed were included with the monsters. Some of the monsters, however, were missing animations that were expected to be there, so it was decided to not make them into **characters**. As for the human characters, the starter animation pack provided by Epic Games was meant for shooters, so it did not fit particularly well with what was needed. After checking out the Maximo character pack, it was discovered that it had the animations that were required for idle, walking and hitting. The problem was that the animations included in the different packs were using different skeletons as seen in Figure 14, because of this the skeletal meshes from Infinity Blade Warriors could not use the animations from the Maximo character and Starter animation pack.

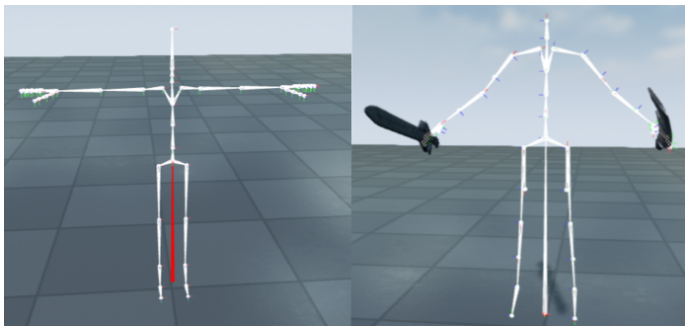


Figure 14: Different Skeletons

6.7.1 Skeleton Re-Targeting

To get around the problem with different skeletons, Persona provides a skeleton re-targeting system. In each skeleton, a humanoid rig is added. The rig serves as a translator between the different skeletons. To use the rig, it is selected in the re-targeting manager, then the skeleton bones are mapped so that they match that of the rig as seen in Figure 15. Once that is done, all the skeletons that use the same rig share all their animations.

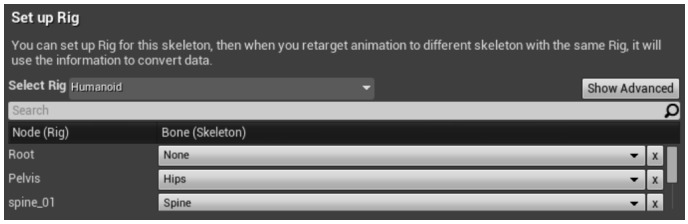


Figure 15: Rigging

6.7.2 Animation Graph

When all the rigging is done, the animations could then be accessed in the Animation Blueprint for each character. In the Animation Blueprint's animation graph, a state machine is created to transition between animations, and output the final animation pose for each frame as seen in Figure 16.

Figure 16: Final Animation Pose

Inside the state machine, all the different animation states that the character can be in are kept, as seen in Figure 17. The states are then transitioned between by checking the translation rules for the current state. If any translation that moves from the current state returns true, the current animation transitions to that animation state.

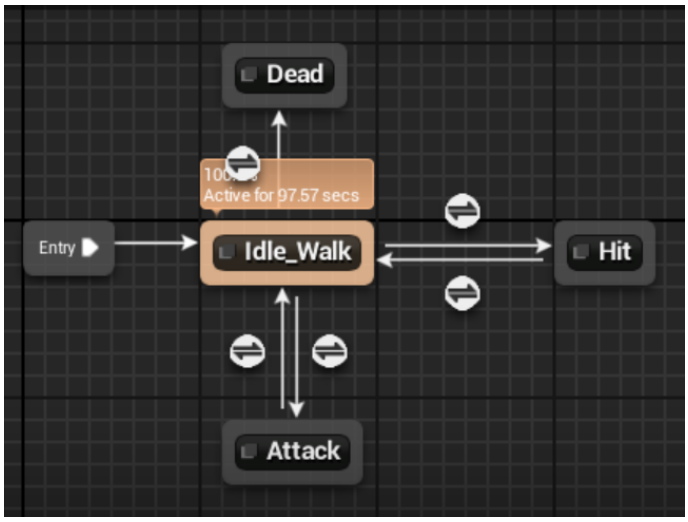


Figure 17: State Machine

6.7.3 Animation Event Graph

In the animation event graph, the animation update event is used to update the transition variables that drive the animations as seen in Figure 18.

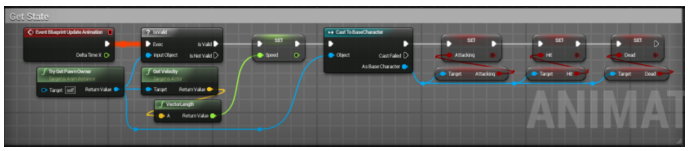


Figure 18: Animation Update

In addition to updating the transition values, animation events are also accepted in the event graph as seen in Figure 19. These events are triggered when an animation has reached a certain point in its time line, and are used to ensure that the animation tells the character that it has completed its animation. Without them, the character would get stuck in that animation, as the transition values would remain the same.

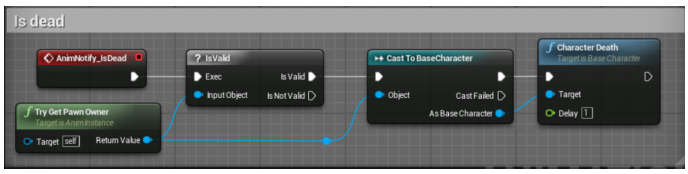


Figure 19: Animation Events

6.7.4 Blend Space

Lastly, one dimensional blend spaces are used to transition between walking and running for characters as seen in Figure 20. This blend space is its own state in the state machine and internally blends between walking and running based on its input parameter. The reason it is one dimensional is because it only takes on parameter, in this case, speed. The speed is then retrieved in the update loop and applied to blend space in the state.

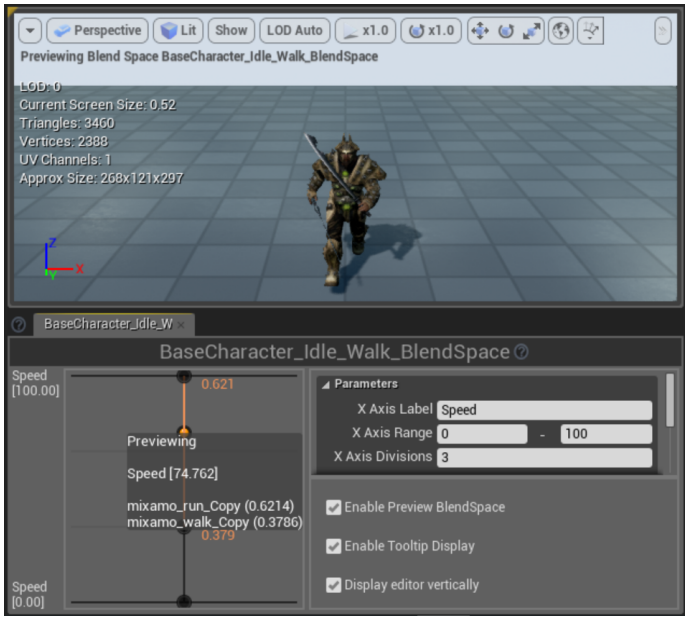


Figure 20: Blend Space 1D

6.8 AI

When it comes to the actors themselves, the difference between the human player and the Artificial Intelligence controlled player is simply that when the AI controlled actors are spawned, the regular "Unit Controller" Blueprint is swapped out with the "Basic AI Controller".

Towards the end, there were some Software issues with Source Tree and Unreal Engine 4 that prevented the "AI Attack" Blueprint from being uploaded properly to the repository prior to the hand in of this thesis, and this Blueprint will therefore not be explained or shown here.

6.8.1 Basic AI Controller Blueprint

This Blueprint is the one that the AI actors have attached to their pawns, and runs the "AI BT Follower", which in turn runs the other AI Blueprints.

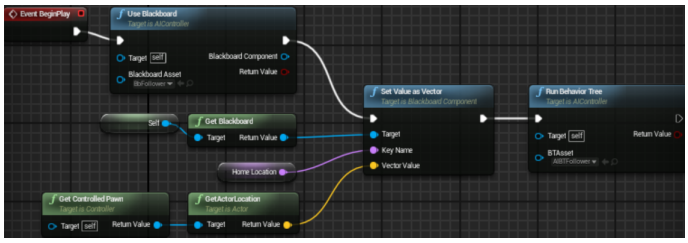


Figure 21: Basic AI Controller Blueprint

6.8.2 AI BT Follower Behaviour Tree

The Behaviour Tree is where everything comes together, by first accessing the "Agro Check" Blueprint, and then move towards the preset location before stopping. There is

an additional attack Blueprint which is accessed after the "Stop When Close" Blueprint, but due to uploading issues, this has been temporarily cut from the Behaviour Tree as well.

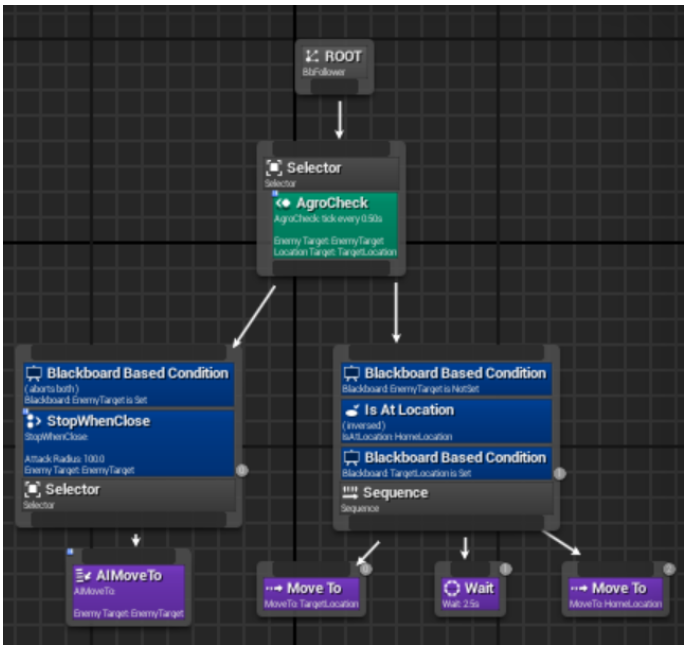


Figure 22: Behaviour Tree

6.8.3 Aggro Check Blueprint

The "Basic AI Controller" itself first checks if the controller reference has been set, and if not, set it, as seen in 23. This is to prevent any overflow of data slowing down the game. After this, the controller starts the check to find all actors on the map that are BP_Base_Characters and are on the same team as the self-actor. This is done by doing a for-loop through all BP_Base_Characters on the map, and then finding the reference to each individual unit. If the reference says that the actor is on the same team as the self-actor, the actor in question is added to an array containing all friendly actors. This can all be seen in figure 24. When the for-loop is completed, the program continues to the "Multi Sphere Trace for Objects" node.

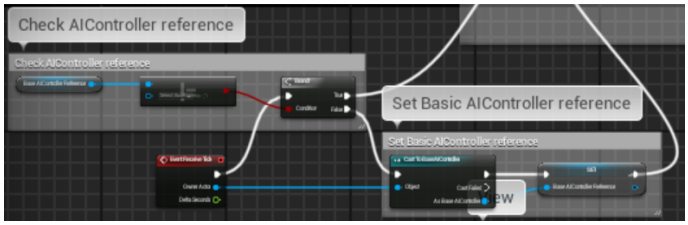


Figure 23: Setting the controller reference

Figure 24: Finds team-members

"Multi Sphere Trace for Objects" then checks all actors it can hit, starting from the self-actors position, but ignoring all non-BP_Base_Characters and all BP_Base_Characters that are on the same team. The program then continues to a for-loop with a break-point, which stops at the first actor that fulfills the requirements. The "Multi Sphere Trace for Objects" and for-loop can be seen in figure 25. Finally, the program moves on to either moving to the actor it wants to attack or the last known location of that actor, as seen in figure 26. Both of these variables are stored in the Blackboard, so that they may be saved and accessed easily by all Blueprints.

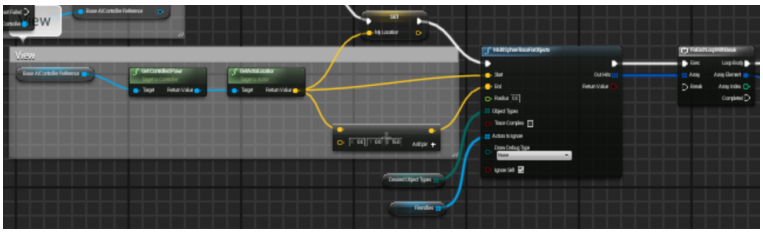


Figure 25: Multi Sphere Trace and for-loop

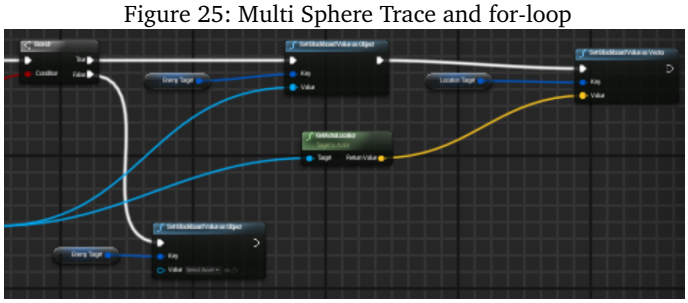


Figure 26: Sets target location

6.8.4 AI Move To Blueprint

This is perhaps the simplest Blueprint, as it only moves the self-actor to the preset position.

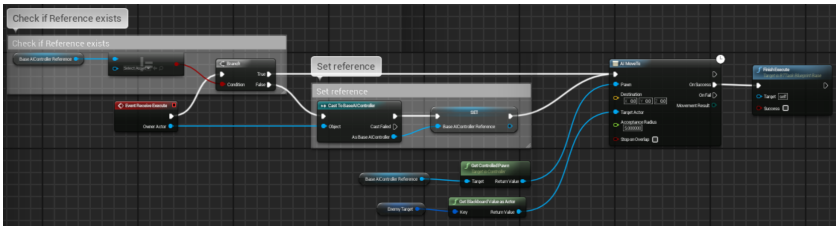


Figure 27: Moves to location

6.8.5 Stop When Close Blueprint

This is a more versatile sub-version of the Basic AI Controller Blueprint, as it will override the other Blueprints and halt the self-actor if it is close enough to the target (target is within the attack radius).

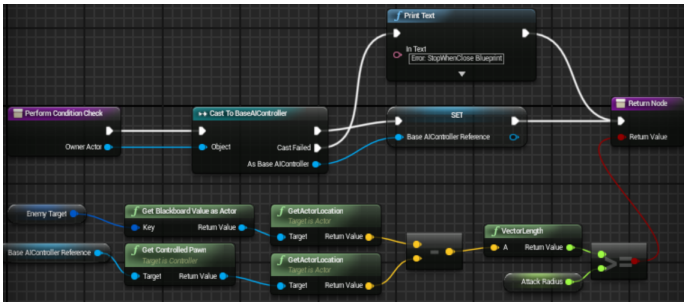


Figure 28: Stop When Close Blueprint

6.9 Level-Streaming

In order to enable level-streaming in the engine, it is first necessary to describe the level composition to it, as can be seen in Figure 29. In the example, the world is composed of a persistent level with two sub-levels. For this game, the persistent level contains most of the terrain and persistent content, while the sub-levels are the dungeons, which contain additional local content and game-play logic.

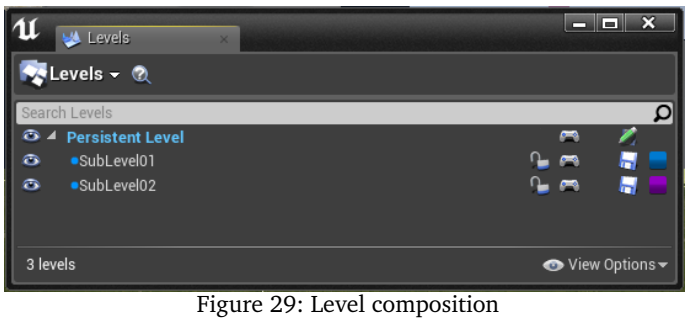


Figure 29: Level composition

Once the composition has been described and the levels have been constructed, it is necessary to add the volumes that will trigger the streaming into the persistent level. These volumes can be seen in Figure 30.

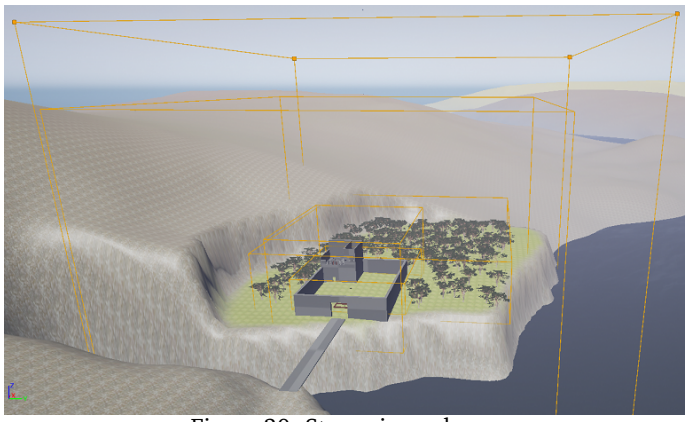


Figure 30: Streaming volumes

To enable levels to cooperate properly with the level streaming, it is necessary to have a global game state. This is because when a level is unloaded, any state it held it reset. However, any changes made inside the level during play will persist between loads. This means the level itself cannot contain flags that indicate whether its initialisation logic has already run. The solution is simple, but not exactly elegant, and involves creating an Actor Blueprint which will encapsulate all of the state variables. This actor, called 'BP Globals' in this project, has no visible model and is instantiated at an arbitrary location inside the persistent level. All the sub-levels' Blueprints then query this actor to determine whether certain logic should execute or not. A snippet showing how such a check is implemented can be seen in 31

Figure 31: Querying global state

6.10 Camera and Mini-Map

The Camera and the Mini-Map are connected, as they are both looking at the same position, but they are implemented in two separate Blueprints. However, in order to ensure that they are properly synchronised, it was decided to have one "master" Blueprint communicate to the other via Custom Events. In this case, the Camera Blueprint is responsible for the synchronisation, both during initialisation and when the position of the camera is updated.

6.10.1 Camera

The Camera was implemented using primarily two components; the Spring-Arm and the Camera Component. The Spring-Arm moves along the terrain and attempts to hold the Camera at a certain height and angle above it, which depends on the player's zoom level. To achieve this, the Camera Blueprint evaluates whether the user is moving the Camera, and if so performs a ray-trace to determine the height of the terrain where the Camera is looking. The Spring-Arm's position is then updated to reflect the change in terrain height, which gently pulls or pushes the Camera with it. An overview of this Blueprint can be seen in Figure 32. For the purposes of readability and management, the logic has been split into multiple sections.

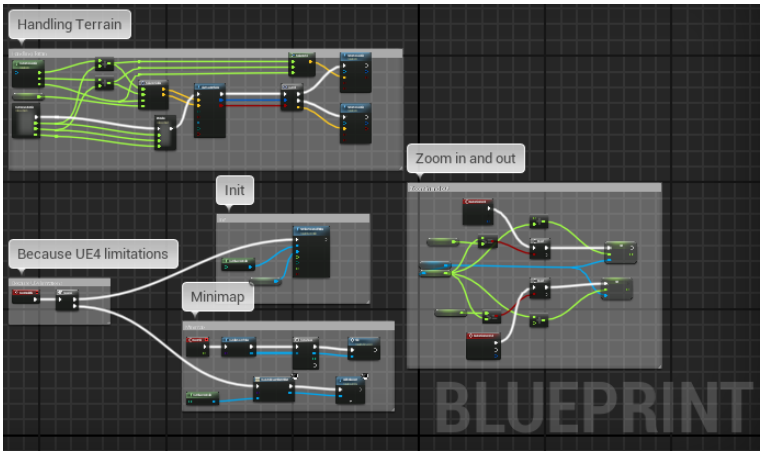


Figure 32: Overview of the Blueprint

A more detailed look at the biggest section, the "Handling Terrain" section, can be seen in Figure 33. This is the section responsible for updating the Spring-Arm's position. A key thing to note here is that in the event that the ray-trace fails, which can happen spuriously, the Blueprint will make the assumption that the previous result is still valid and fall back to it. These failures generally happen near the edges of a terrain section.

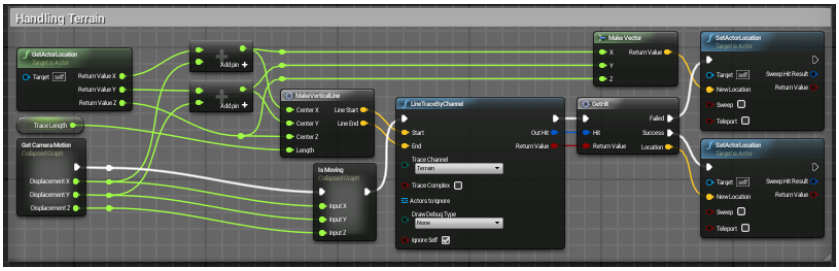


Figure 33: Detailed look at the "Handling Terrain" section

6.10.2 Synchronisation

The synchronisation with and initialisation of the Mini-Map from within the Camera Blueprint is shown in detail in Figure 34. At initialisation, the UI widget is created and connected to the player's viewport. The other part of the synchronisation is the update of position, which occurs at each engine tick. During these ticks, the Camera Blueprint finds the Mini-Map instance and invokes its Tick Custom Event.

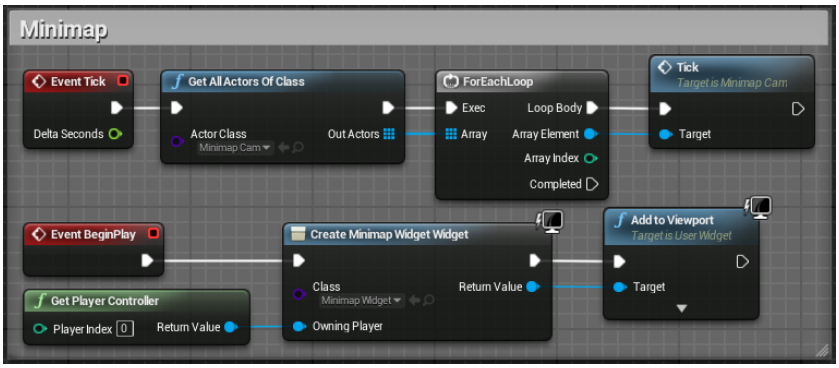


Figure 34: Mini-Map synchronisation and initialisation

6.10.3 Mini-Map

On the receiving end of the Tick Custom Event is the Mini-Map Blueprint, which as Figure 35 shows, moves the Mini-Map's Scene Capture Component so that it remains in sync with the Camera.

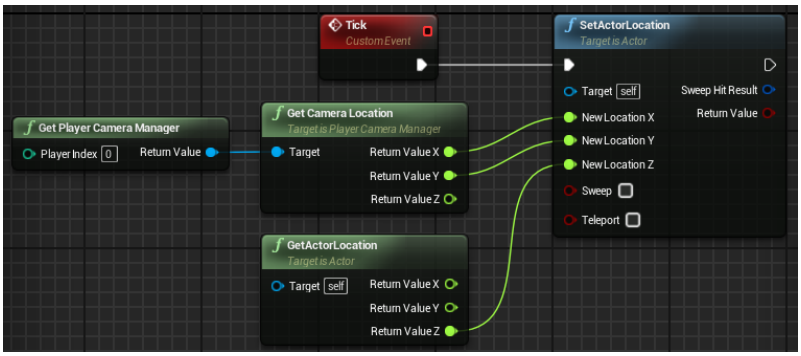


Figure 35: Mini-Map Blueprint

One of the key differences between the Mini-Map and the Camera is that it does not use a Spring-Arm, and instead hovers at a constant height. It also does not render directly to the player's screen, but rather to a texture via the Scene Capture Component. Although it isn't clear from any of the Blueprints, this is the texture utilised by the Mini-Map Widget, and this behaviour is controlled from the settings of the individual components.

6.11 Procedural Foliage

For the procedural foliage, multiple components are needed in order to fulfill the different requirements for generating it. Most important of these is the terrain material Blueprint, which is responsible for blending different terrain textures. Based on this blending, it can decide where different types of foliage may be placed. The Blueprint for this is shown in Figure 36. The two nodes seen at the bottom-left will sample the terrain and mark appropriate areas for the engine to spawn the grass foliage, which in this case is where the texture on the terrain is grass and not rock. This could be easily extended to add rocks, plants and other types of grass. The result of applying this grass on a patch of terrain can be seen in Figure 37.

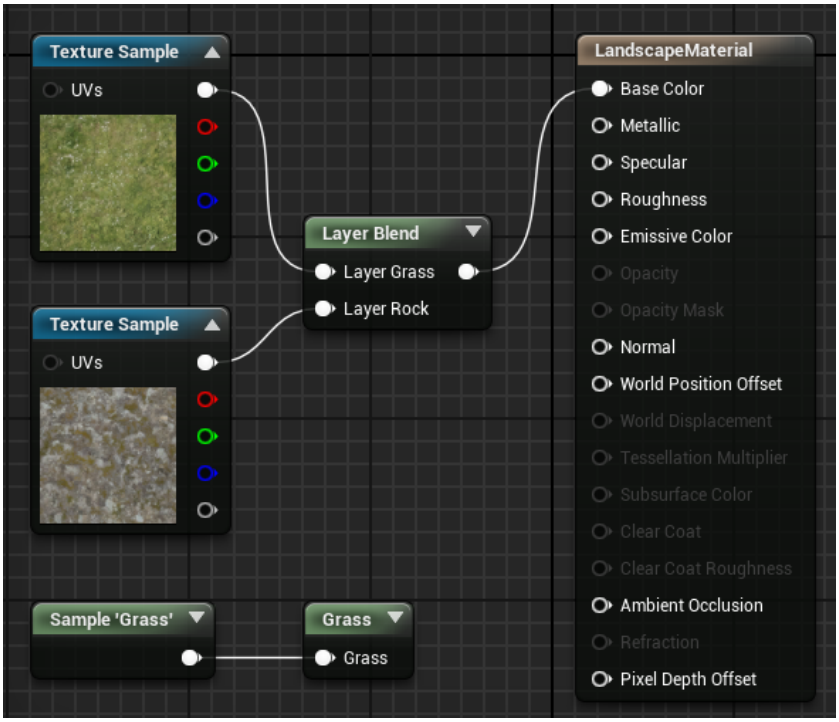


Figure 36: The landscape material Blueprint

For the trees, and potentially other sparse flora, Procedural Foliage Volumes and Procedural Foliage Blocking Volumes are used to define the areas where the growth simulation takes place. In order to use these, it must first be enabled in the editor - as it is an experimental feature - under Editor Preferences > Experimental, shown in Figure 38. The number of generations to simulate, growth rate and shade radius are adjusted in the foliage's settings to produce the desired effect. The settings used in this project can be seen in Figure 39.

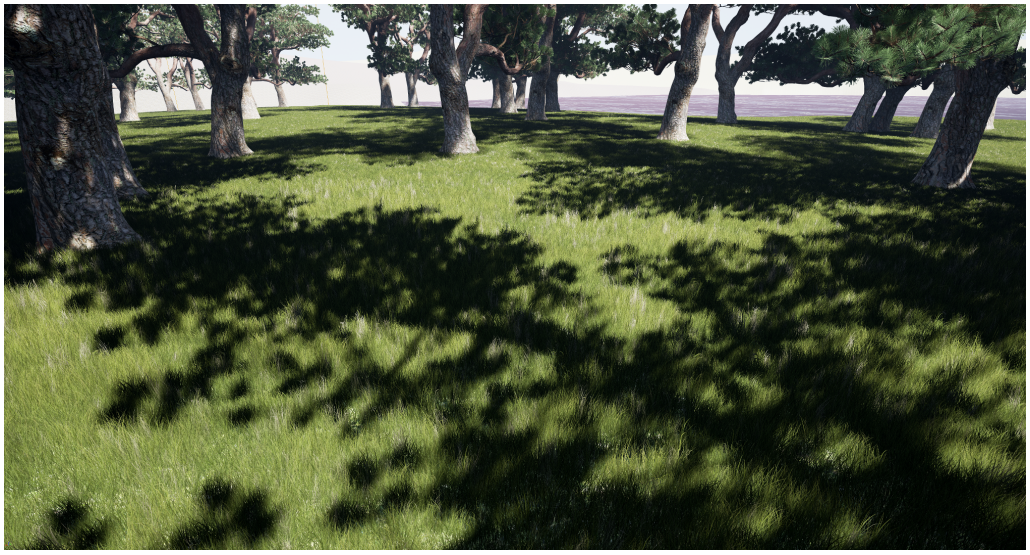


Figure 37: A patch of grass that has been procedurally generated from the Blueprint

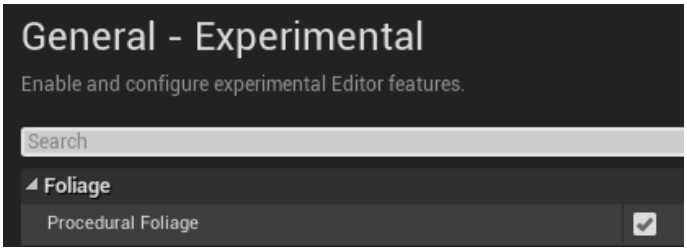


Figure 38: Enabling the use of procedural foliage generation in the Unreal Editor

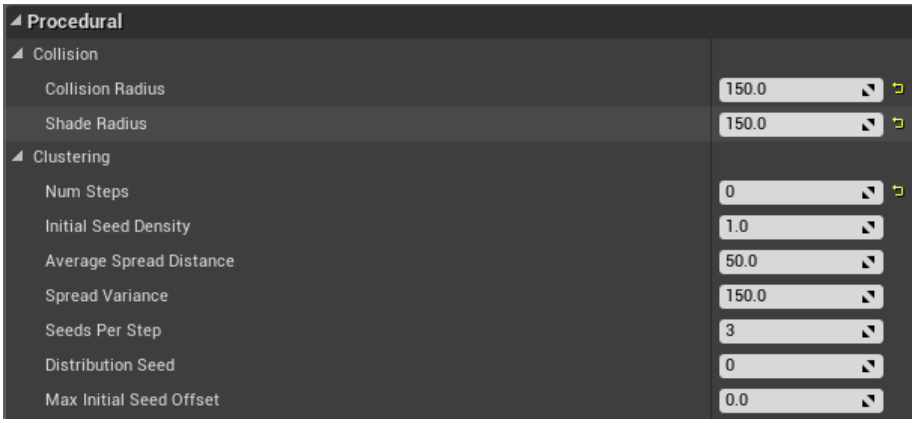


Figure 39: The selected settings for the generation of procedural trees

6.12 Lighting and Shadows

Due to the massive scale of the map, and the limited RAM on most of our computers, using static lighting for this project was not feasible. Calculating the lighting would usually stall the computer for minutes, then forcing the operating system to close the editor because it consumed all available RAM, and finally crashing the driver due to faulty code in the editor. Occasionally, the editor would also cause a Blue Screen of Death for unknown reasons while static lighting was enabled. This is appears to be a shortcoming of the editor's design not considering workstations that have less than 32GB of RAM, thus being unable to respond to issues related to memory constraints.

For these reasons, dynamic lighting was used instead. However, the default dynamic lighting implementation in the Unreal Engine is not only of lower quality than the static implementation, but also has a big performance impact during game-play. This is because the dynamic lighting is calculated in real-time, instead of being a look-up into precomputed shadow-maps. There are, however, certain experimental features that can be enabled to make dynamic lighting look better than the static version, specifically Distance-Field Ambient Occlusion, Ray-Traced Distance-Field Shadows and Height-Field Global Illumination. These features mostly influence the build-time and have good real-time performance. Additionally, because dynamic lighting is used instead, it is possible to get realistic time-of-day changes with no additional computational overhead.

Finally, to ensure the scene does not become too dark, and to help mask the hard shadows, a dome light is added to the scene which represents the lighting contribution from the sky. This dome light is a texture-based light which captures the sky and large surrounding mountains, or other influential objects, in order to produce a realistic effect. Figure 40 shows the results of including the lighting contribution from the sky and enabling distance-field ambient occlusion on some trees. The distance-field ambient occlusion adds considerable depth and detail to the geometry, while the dome light adds the light that one would expect from the sky in the real world. Without this light source, it would appear as if there was only one strong light in the scene, which is unrealistic and sticks out to humans.

In order to enable these features, three primary steps have to be taken. Firstly, under Project Settings > Rendering, the option "Allow Static Lighting" must be un-ticked, while "Generate Mesh Distance Fields" and "Generate Landscape Real-Time GI Data" must be ticked, as can be seen in Figure 41. Secondly, in the assets that should be affected, the setting "Cast Static Shadow" must be un-ticked, while the settings "Cast Shadow", "Affect Distance Field Lighting" and "Cast Dynamic Shadow" must be ticked, as can be seen in Figure 42. Finally, to take advantage of dynamic time-of-day, a Blueprint must be written to update the directional light-source in the scene depending on the calculated time-of-day. To enhance this effect, a realistic texture should be used for the skydome as well, which should be updated along with the light-source. For the skydome it made sense to reuse the free asset created by Epic Games for their Kite Demo, named 'BP_Sky_Sphere', as it had both a realistic texture and straight-forward integration with the directional light-source. Figure 43 shows the settings applied to the skydome after placing it in the world, and Figure 44 shows the Blueprint interacting with it.

Figure 40: The top image has both the dome light and distance-field ambient occlusion enabled. The middle image has the dome light on, but the distance-field ambient occlusion off. The bottom image has both the dome light and distance-field ambient occlusion disabled.

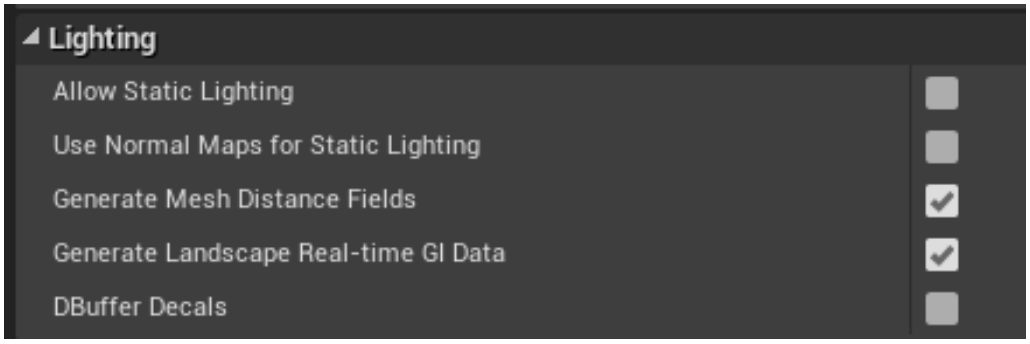


Figure 41: Settings for enabling Distance-Field based features, as well as Height-Field Global Illumination.

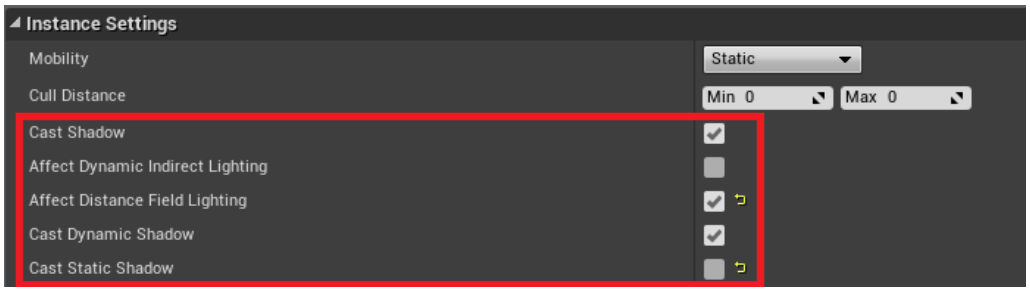


Figure 42: Settings in the asset specifying its interaction with the lighting.

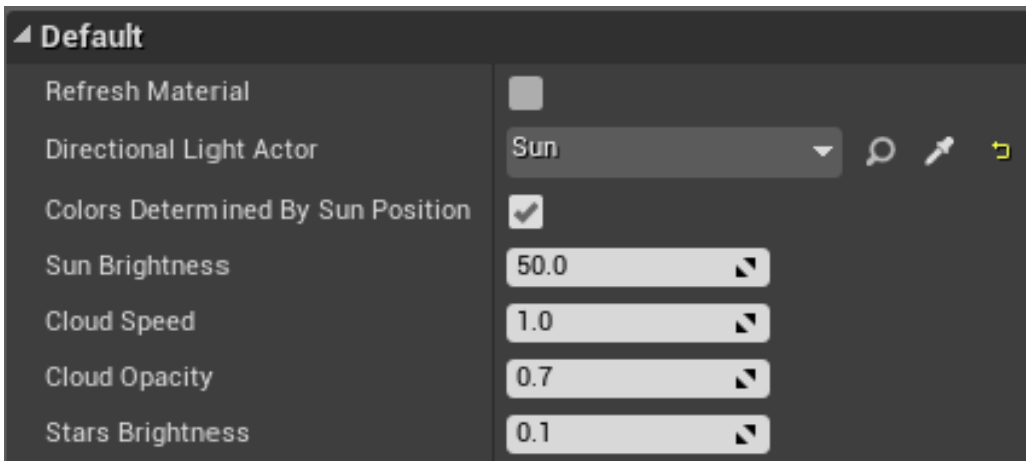


Figure 43: BP_Sky_Sphere instance settings, connecting it to the directional light-source in the scene - called 'Sun' in our project.

Figure 44: Blueprint adjusting the time of day

7 Deployment

Where the project and game can be found and how to install and play the game on your computer. The Google Drive folder containing the executable and the .zip file are both found here:[12]. Link to game-play video can be found here:[13].

7.1 Executable

An executable version for Microsoft Windows 7 / 8 / 8.1 / 10 will be publicly available for some time here:[14].

7.2 Building

To build the game it is recommended to use Unreal's packaging options as the authors did on the game project. An executable version for the game is then automatically made in a folder along with any needed assets.

A project zip will be hosted by the authors for some time under this link:[15]. Unreal Engine 4.10[16] is required to run the project. It is recommended that you download the project zip, then build the game using the Unreal editor as mentioned above.

Although all of the code and Blueprints are hosted on Bitbucket, there is not enough space in the repository to host all of the required assets. Therefore, downloading the repository will not yield a complete, buildable project. The Bitbucket repository can be found here:[17].

7.2.1 Problems

This section describes problems that the authors noted can be encountered during packaging.

Git

Some problems with Git while downloading the repository may be encountered, but this varies between individual setups. Primarily these problems will be with the connection being reset, or indexes being corrupted. A good, stable Internet connection should resolve these.

OS

While it is possible to build the project for Windows and Linux with a Windows OS, it is not possible to build the project for Mac without a Mac OS.

Path Length

In some cases on Windows, the path length of some files may exceed the limit of 255 characters allowed depending on where the project is on the disk. It is recommended that the project is not stored too deep within a disk, as this will cause errors when packaging the project. The project can still be opened, however, and if necessary, the folders' names can be shortened from within the editor in order to circumvent this issue.

8 Testing

This chapter contains all the testing that was done during development, including the goals, the results, and what was learned from tests on the authors' own machines, as well as student testing, feedback and response.

8.1 Testing with People

While testing, they were allowed to play the game as long as they wanted. After playing the game they were asked to fill out a survey, which may be found in Appendix L. The goal of the survey was to gather information about the following aspects of the game:

- Design of User Interface
- How intuitive is the User Interface
- Playability of the game mechanics
- Potentiality of multiplayer integration
- Graphics and aesthetics
- Balance of the game-play, game mechanics and representation of the world
- Controls and responsiveness
- Interest in the game as a whole

The information gathered in this survey has been of great use, as it depicts what has been done correctly, what needs to be worked on, and if the game is an idea worthy to be expanded upon in the future. The results of the survey can be found in Appendix M.

8.1.1 Feedback

This sections contains the feedback from the survey above, which was answered by 4 different people. The initial goal was to get more testers, but this did not happen because the survey and tests were not ready until late in development. The main reason for this was the low hardware performance of the available laptops, so testing on campus was problematic. Although a powerful desktop computer was used through Windows' Remote Desktop, there would occasionally be high ping - up to 1000 ms - which would've rendered the test invalid. As a result, the demo was sent to the testers individually.

UI

- **Main menu's** design received a low score, but the authors would argue that this score is caused by the lack of original textures, and once completed the menu would be satisfactory.
- **Settings menu's** design is more well-liked.
- **Game selection menu's** design is, again, OK, but somewhat lacking. The authors would again argue that the low score in design is because of the lack of original textures.
- **HUD's** design is fine, but needs general improvement.

Generally, the sub-menus under the Main Menu have been rated as very intuitive, most

likely because of their simplistic or familiar layout, while the in-game HUD seems to need some improvements to increase its user-friendliness.

Controls

The controls were OK, fine tuning is necessary - as one would expect - but there are otherwise no serious flaws that need to be addressed. The camera seems to be the main area to work on, like increasing its speed and adding Quality of Life improvements, such as enabling it to be pushed by moving the cursor towards the border of the screen. It also needs to be made smoother and more responsive.

Balance

Balancing is a long and meticulous process and, as expected, the game is not quite at the point where it feels balanced. This was made clear from the survey.

Game Play

Currently, it would seem that the difficulty of the game is too low. The testers did, however, find the mechanics of the game easy to understand, as they had all played similar games before. The fun-rating of the game seems to indicate that it is rather bland, but this was expected, as there are no hero units or abilities to use. Information about each unit should also be increased, and the possibility to allow the player to see more information about the enemy team should be considered.

Multiplayer

The option to include multiplayer in the future is a subject for discussion, as opinions are divided. There would need to be more testing done to get a more definitive answer, and it may have been clearer had the game been further in development.

8.1.2 Working with M.Sc student Amund Lieng

During the project we requisitioned the help from Amund Lieng, a Masters degree student at NTNU Gjøvik developing a testing heuristic for games, with some high-quality play-testing. Through his testing and analysis, he found several areas that needed improvement or adjustment - some of which were, in fact, unknown to us prior to his assistance. The following list of issues was compiled from e-mails he sent:

- Fix camera position resetting.
- Make arrow keys also fire same input as WASD.
- Make the keybindings visible even if not implemented.
- Middle mouse move camera.
- Check end game condition not only on end turn.
- More information when it's your turn.
- Indicate no more action points.
- Move and attack in one action.
- Red path when none is valid
- Better indication of friend and foe
- Highlight enemies within range.
- Fix spelling errors and inconsistency.
- Fix difficulty as it has no effect.
- Select ring under monsters are hard to see.

- Move difficulty to the other side of the image in skirmish and campaign menu.
- Better messaging when starting without choosing a level.
- Background on mouse over information as it is hard to see over some textures.

A bulk of these issues were resolved during development, as this testing and feedback was done as part of the middle phase. However, some of the issues persist because of implementation difficulty or because they were minor issues that were not prioritised.

8.2 Balancing

Balancing the game is used to make sure that the game is fair and improve the user experience.

Initially, the weapons and units had been given statistics that the authors thought matched. While the authors were testing, some imbalance became obvious.

8.2.1 Characters

Initially, damage was increased using the character's attribute that matched that of the weapon. So if the weapon was a strength weapon, the character's strength would be added to the damage with a multiplier. This was done away with, as it made the weapons difficult to balance between each other, and increased the damage of the most powerful characters too much. The speed of some unit was also making them unable to do much each turn so their speed was increased. In addition hit chance was also increased. The complete changes to the statistics for the characters can be found in Appendix P.

8.2.2 Weapons

The weapons' attack ranges were initially too high when considering the reach of each character, and the attack cost of each weapon needed to match the damage dealt and modifiers. The changes from the original values can be seen first in Figure 45, and the result of balancing is in Figure 46, although it should be pointed out that the game still requires balancing. The full statistics changes for the weapons can be found in Appendix O.

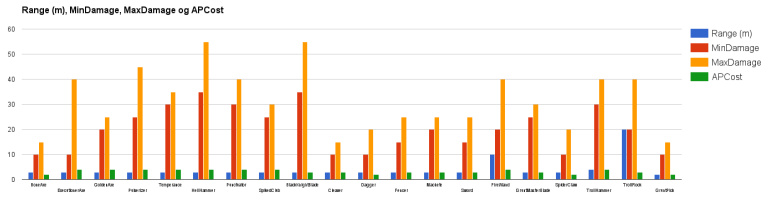


Figure 45: Starting weapon statistics graph

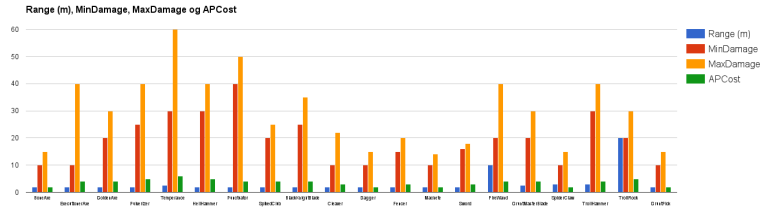


Figure 46: Rebalanced weapon statistics graph

8.2.3 First attack

Another issue is that the team which starts the match has an advantage, since they get to attack first. As the whole team moves at once, the team which goes last will most likely lose some characters before it is their turn. To compensate, the starting team does not get action points regenerated at the start of there first turn, but the other team does, allowing them to catch up once its there turn. Another way to get around this dilemma would be to spawn the characters from each team further away from each other, thereby giving each team time to carefully position their units to gain an advantage.

8.2.4 Difficulty

To balance the difficulty, each separate difficulty spawns different characters. This was done to avoid having to balance each unit's statistics at different difficulties, or instead of having to make the enemy team the same unit but with more health or attack power. Instead, each difficulty can be balanced by giving the opponent better units or more characters at higher difficulties. This also increases the playability of each level, as each difficulty gives a different experience.

8.3 Performance and Reliability

For measuring performance within the game, Unreal Engine provides two statistics generation commands. While playing the game, the command 'StartFPSChart' is written in the console, which will make the engine begin gathering statistics. The gathering is then stopped with the command 'StopFPSChart'. Once stopped the engine compiles and outputs the gathered statistics. Charts made from these statistics can be found in Appendix K. It should be noted that for this test the highest settings were used, with a resolution of 1920x1080.

During the performance tests, the specifications listed in Table 1 were used. Results may

be different with other configurations.

Operating System:	Windows 10 64-bit
CPU:	Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz
GPU:	NVIDIA GeForce GTX 780
Resolution Quality:	100%
View Distance Quality:	3
Anti-Aliasing Quality:	3
Shadow Quality:	3
Post-Process Quality:	3
Texture Quality:	3
Effects Quality:	3

Table 1: Hardware specifications and game settings used during performance tests. In Unreal Engine, a quality setting of '3' is also known as "Epic", and is the highest setting available.

In addition to the statistics in Appendix K, the engine also outputs the frame rates, and these can be seen in Table 2.

Map	Average FPS	Time over 30 FPS	Time over 60 FPS
Long Meadow	115.98	100%	100%
Brittle Bridge	119.15	100%	99.65%
Persistent Level(Campaign)	66.61	100%	96.70%
Main Menu	85.27	71.61%	71.12%

Table 2: Frames per second during testing, and the time spent over certain thresholds

Although the FPS on the Main Menu is anomalously low, the actual rendering of each frame is in fact fastest there, as can be seen from the statistics in Appendix K. It may be that the engine skips some frames, possibly due some optimisation since the menu is mostly static, which could explain these results.

While testing the reliability of the game, the authors were able to have the game running for at least 5 hours without experiencing any crashes. The games modes have also been played through successfully without the game crashing.

9 Discussion

This chapter is used to talk about our interpretations of the process. It is a discussion of the evolution of our work, its flaws, what we have learned and what the game we have made may be expanded to become in the future.

9.1 Results

At the outline of the project, we always had creating a playable game as our main goal. It was therefore important for us to establish a clear game design document, and try to establish a high-level plan of the game, which could be used to guide our work throughout the semester. Despite the large number of features that were left unimplemented, we believe we have achieved our initial goal, as we can see that there is a clear potential in the game and its mechanics. Actual further expansion on this specific project is, however, unlikely, as we would instead like to start from scratch with the fresh knowledge we have gained if we were to make a game that would eventually be put on sale.

9.1.1 Major issues faced during development

Unreal Engine

RAM requirements was an issue we had not faced prior to this project, and proved to be difficult to work around. Some of the textures had a resolution of 1024x1024 or higher, with some textures we found going all the way up to 8192x8192 resolution. This made it difficult to test them at times, as Unreal Engine 4 assumes all computers attempting to access such textures have 32GB RAM, rather than the original 8GB RAM minimum requirement for normal use of the engine.

Bitbucket

With a repository cap at 2GB, Bitbucket filled up quickly with the textures and meshes we needed for the project, and several of these eventually had to be added to the git-ignore file and shared manually through Google Drive instead.

9.1.2 Learning experience

Using Unreal Engine 4 thought us much on how to use tools that are similar to programming, yet different, here in the form of Blueprints. It has also thought us more on how to quickly adapt to new environments, as none of us had any prior experience with the engine, and how to solve issues with little documentation on the subject, as Unreal Engine 4 updates so frequently that documentation on similar issues is quickly outdated.

We have had some problems with branch merging, as certain merges came later than expected, as well as branches becoming larger and more complex than they should have. This has resulted in several major merge-conflicts when trying to bring everything together, and slowed down project work later in the process. This has therefore been a valuable lesson to merge much more often, preferably once a week, in order to avoid such major conflicts.

9.2 Future Work

Many of the original planned features did not make it into the final game, and more potential features and improvements were discovered during development. These are the main ones.

9.2.1 Features

Campaign story

Even in the beginning of the process, the story for the Campaign and the game name were down-prioritized, as the game-play itself was considered more important. However, at the end of the project, "Kingdom: Civil War" was suggested as a name for the game, and a story built around the concept of several factions within a kingdom fighting for supremacy may be a logical tie-in to the game as it stands now, as larger armies were cut in favor of single unit/ hero combat.

Conquest

Future Conquest mode would include the ability to choose between several different maps, rather than just the current Campaign map, with the possible addition of randomly generated maps. This mode is explained further in the 4.1.1 Initial Design section.

Instance/ Skirmish

Currently the skirmish game mode only supports one type of victory condition. Adding different game modes to the skirmish maps such as defending a certain character, holding a position for an amount of turns or capturing a point is certainly an option for the future.

Buildings

Buildings had to be cut entirely from the game due to time-constraints, lack of textures and a general low priority for this feature on the whole. Some grey buildings have been added, but they are there only as examples of what it might look like to have them in the Combat-phase of the game, and serve no purpose. More details on buildings may be found in the 4.1.1 Initial Design section.

Factions

One of our earlier ideas was to implement factions, either in the form of humans versus monsters or one type of human/ monster alliance versus another. This had to be cut in the game, as we did not have time to balance two or more factions, despite the implementation of this on a low-level "lock certain units for each faction"-style being simple enough to be put in the game.

Units and Heroes

Our current layout of the game makes no distinction between regular units and heroic units, as unit and hero production is not yet implemented. In an expanded game, these would be more distinguished, and would become more similar to what has been explained in the 4.1.1 Initial Design section.

Multiplayer

Despite not being a subject of much focus neither at the start of the project nor during development, multiplayer compliments the turn-based game-play style nicely, and

both local multiplayer, where for example controller support may be added, and online multiplayer support may be considered for future implementation.

9.2.2 Improvements

Graphics

While most necessary graphics have been included, they have a large impact on performance for computers that have not been built specifically for gaming. The main improvement to be considered for graphics is, therefore, the ability to tune the graphics settings in the graphics menu. Further improvement would be to include more variety in foliage, add rocks and boulders, more terrain textures, particle effects and more tweaking of the fog and lighting. It could also be worth investigating getting static lighting to work on weaker computers, and disable dynamic time-of-day and the other GI features in favour of increased performance on those platforms.

Menus and options

The menus only use textures and images that the authors made or were included in Unreal. It would be preferred to have a menu style that was uniform. In addition, the layout of the menus both in the main menu and in game could certainly be improved to be more aesthetically pleasing.

The options menu are quite lacking. Key-bindings would need to be implemented and graphical options added. In addition game play options for implemented features, an example would be disable and enable sub titles.

Units

The current units all use a pre-made set of skins and animations, and is therefore preventing any sale of the game as these skins and animations are not made by us nor are very unique, as they come free in the Unreal Marketplace. All units would therefore require brand new skins and animations in any expansion of the game, which may prove difficult, as none of us have any experience or education in 3D-modelling or animation.

AI

The Artificial Intelligence present in the game is currently underdeveloped, and has several features missing from it. These include:

- Over-world AI
  - The current AI is limited to the combat sequence, and will therefore not perform any actions on the Campaign/ Conquest map.
  - This extension would also include strategic decisions, defensive and offensive capabilities for the AI, as well as the ability to use the same tools as the human players.
- Improved AI co-operation
  - Each AI controlled character currently acts independently and only goes for enemies on their own.
  - AI co-operation would make them protect their wounded and perform coordinated group attacks on strong enemies.

- Difficulty implementation and settings
    - Easy difficulty with several random decisions and poor judgment calls.
    - Medium difficulty with some random decisions and occasional bad judgment.
    - Hard difficulty will play as perfectly as possible. This is the one that has been (attempted to be) implemented.

10 Conclusion

Although there are several features that had to be left out, we feel that the game itself not only shows potential, but also functions as a playable game, albeit on a smaller scale than originally intended. We can see that as it is now, it may not be enjoyable in the long run, but can be fun to play for 1-2 hours instead, even more so if local multiplayer where the players switch between turns is used. Added multiplayer may therefore have been a true focus for future implementation, both locally and online, as it seems to be enjoyable.

Unreal Engine 4 is a big engine, and has become an industry standard. It was therefore very beneficial for us to learn this tool while working on our project, not only if we should want to create our own engine, but also if we are going to collaborate with other people on larger projects in the future. Unreal Engine's visual scripting language Blueprint should make working with designers, artists and other non-programmers easier for us as well, and it is a great tool for rapid prototyping. It is also a good long-term investment to learn the Unreal Engine, as the engine and its community will only continue to grow for the foreseeable future.

Overall, we are proud of what we have made, but more importantly this has been a great learning experience that will be useful for future work. We have faced several challenges throughout this project, and by overcoming them we have gained new insight into the process of making games, as well as additional confidence in our abilities to work together as a group towards a common goal.

Bibliography

[1] Epic Games Inc. *Hardware and Software Specifications for Unreal Engine 4*. Accessed 04. May 2016. URL: https://docs.unrealengine.com/latest/INT/GettingStarted/RecommendedSpecifications/.

[2] *Scrum*. Accessed 04. May 2016. URL: https://en.wikipedia.org/w/index.php?title=Scrum_(software_development)&oldid=718741607.

[3] Epic Games Inc. *Unreal Engine Coding Standard*. Accessed 16. May 2016. URL: https://docs.unrealengine.com/latest/INT/Programming/Development/CodingStandard/index.html.

[4] Totally Unreal. *AI video tutorial series, part 1*. Accessed 10. May 2016. URL: https://www.youtube.com/watch?v=4H_1skacmXA.

[5] Totally Unreal. *AI video tutorial series, part 2*. Accessed 10. May 2016. URL: https://www.youtube.com/watch?v=Z18gmak2p74.

[6] Totally Unreal. *AI video tutorial series, part 3*. Accessed 10. May 2016. URL: https://www.youtube.com/watch?v=9sW-LQlTJHk.

[7] Epic Games Inc. *AI and Behaviour Tree tutorial*. Accessed 10. May 2016. URL: https://docs.unrealengine.com/latest/INT/Gameplay/AI/.

[8] Epic Games Inc. *Blueprint Behaviour Tree Tutorial*. Accessed 10. May 2016. URL: https://wiki.unrealengine.com/Blueprint_Behavior_Tree_Tutorial.

[9] Epic Games Inc. *Procedural Foliage Tool Quick Start*. Accessed 10. May 2016. URL: https://docs.unrealengine.com/latest/INT/Engine/OpenWorldTools/ProceduralFoliage/QuickStart/.

[10] Epic Games Inc. *Grass Tool Quick Start*. Accessed 10. May 2016. URL: https://docs.unrealengine.com/latest/INT/Engine/OpenWorldTools/Grass/QuickStart/index.html.

[11] Epic Games Inc. *Unreal Engine Persona*. Accessed 16. May 2016. URL: https://docs.unrealengine.com/latest/INT/Engine/Animation/Persona/index.html.

[12] noMemory. *Google drive folder*. URL: https://drive.google.com/folderview?id=0B_0MF6YOU3emMFRHbGZJcnNIdUE&usp=sharing.

[13] noMemory. *Gameplay video*. URL: https://youtu.be/YvVHj1c3cKU.

[14] noMemory. *Game Executable*. URL: https://drive.google.com/file/d/0B_0MF6YOU3emM19NU3k5cnl5VUU/view?usp=sharing.

[15] noMemory. *Project zip file*. URL: https://drive.google.com/file/d/0B_0MF6YOU3emM0V2eXF0cXhKM0U/view?usp=sharing.

[16] Epic Games Inc. *Unreal Engine Download*. Accessed 13. May 2016. URL: https://www.unrealengine.com/what-is-unreal-engine-4.

[17] noMemory. *Online Repository*. URL: https://bitbucket.org/Chromekard/nomemory/overview.

[18] *Frame rate*. Accessed 10. May 2016. URL: https://en.wikipedia.org/w/index.php?title=Frame_rate&oldid=714359649.

[19] *Global Illumination*. Accessed 10. May 2016. URL: https://en.wikipedia.org/w/index.php?title=Global_illumination&oldid=710346677.

[20] *Ambient Occlussion*. Accessed 10. May 2016. URL: https://en.wikipedia.org/w/index.php?title=Ambient_occlusion&oldid=710436040.

[21] *Viewport*. Accessed 10. May 2016. URL: https://en.wikipedia.org/w/index.php?title=Viewport&oldid=705561800.

[22] *Signed Distance Function*. Accessed 06. May 2016. URL: https://en.wikipedia.org/w/index.php?title=Signed_distance_function&oldid=717088902.

[23] *Skybox (video games)*. Accessed 11. May 2016. URL: https://en.wikipedia.org/w/index.php?title=Skybox_(video_games)&oldid=698434422.

[24] *Lord of the Rings: Battle for Middle Earth*. Accessed 06. May 2016. URL: https://en.wikipedia.org/w/index.php?title=The_Lord_of_the_Rings:_The_Battle_for_Middle-earth_II&oldid=719233099.

[25] Electronic Arts Inc. *Lord of the Rings: Battle for Middle Earth*. Accessed 06. May 2016. URL: http://www.ea.com/lotr-the-battle-for-middle-earth-2.

[26] Jacob Penner. *RPGSystems Statistics Playlist*. Accessed 16. May 2016. URL: https://www.youtube.com/playlist?list=PL9v2wOnRXtUpJGOY2tTa8P5dSal9s0QSJ.

[27] Epic Games Inc. *Animation Retargeting on Different skeletons*. Accessed 16. May 2016. URL: https://www.youtube.com/watch?v=xy9aLbZLdeA.

[28] Tom Looman. *Multi-color Outline*. Accessed 16. May 2016. URL: http://www.tomlooman.com/multi-color-outline-post-process-in-unreal-engine-4/.

[29] Jeff Farris. *Damage in Unreal Engine*. Accessed 16. May 2016. URL: https://www.unrealengine.com/blog/damage-in-ue4.

[30] G-Rath. *Custom Blueprint Node Creation*. Accessed 16. May 2016. URL: https://wiki.unrealengine.com/Custom_Blueprint_Node_Creation#Extending_Classes.

[31] Rama. *Enums For Both C++ and BP*. Accessed 16. May 2016. URL: https://wiki.unrealengine.com/Enums_For_Both_C%5C%2B%5C%2B_and_BP.

[32] Rama. *Blueprint Implementable Events*. Accessed 16. May 2016. URL: https://wiki.unrealengine.com/Blueprints,_Empower_Your_Entire_Team_With_BlueprintImplementableEvent.

A Terminology

- **Unreal Engine 4.10** - game engine created by Epic Games Inc., ©2004-2016. All rights reserved. See bibliography link: [1].
- **GT** - Game Thread, a thread on the CPU running the main game loop and related processing.
- **RT** - Render Thread, a thread on the CPU that prepares the commands to be sent to the GPU for processing.
- **GPU** - Graphics Processing Unit, a discrete, massively parallel computation unit specialised for graphics processing.
- **Scrum** - see bibliography link [2].
- **FPS** - frames per second [18].
- **GI** - Global Illumination, a term for techniques that attempt to realistically simulate multiple indirect light bounces [19].
- **AO** - Ambient Occlusion, a technique for masking areas that should not be affected by ambient lighting in the scene [20].
- **Viewport** - The player's view into the game-world [21].
- **Distance-Field** - A 2D or 3D grid of cells that contain a signed distance to the nearest surface [22].
- **Player** - may be either human controlled player or AI controlled player, unless otherwise specified.
- **Skydome** - A textured sphere that surrounds the game-world, which represents the clouds, atmosphere, sun and stars [23].
- **Lord of the Rings: Battle for Middle Earth©** - video game released by Electronic Arts, Inc in 2006 [24, 25]. All rights reserved.

B Tutorials

The following tutorials have been used by the authors are found in the Bibliography under these links: [4–10, 26–32].

C Project Plan

The following pages are the PDF output of our original Project Plan. Certain goals and milestones were altered during development and not all features were used, while other features that are not mentioned here were created during the development of the project.

Project plan NoMemory 13HBSPA

Goal and frame

Background

For this project we were interested in finding and exploring combinations of game genres and mechanics that we had not previously seen, in order to determine if this could result in a game with interesting and fun gameplay.

The combination we went with therefore, was an amalgamation of various gameplay that each group member enjoyed - focusing on that which in theory could synergise well - thus providing a unique and fulfilling experience. Such a design should also ensure that each group member has something they want to pursue.

In short, the design we have in mind is largely divided into two major pieces; a strategic "overworld" mode which draws from the two "Lord of the Rings" games "Battle for Middle Earth" and "War of the Ring", and an action-adventure "dungeon" mode where the player controls a hero who is accompanied by a legion of soldiers from the overworld.

There are, however, mechanics that are as of yet undecided which we intend to resolve over the course of the project. Details of this will be in the final thesis.

Project goals

Result goals

- Have a running prototype of a game demonstrating unique mechanics and/or storytelling.
    - Good communication between group members and well documented code, providing understanding of everyone's code at any given time.
    - A clear cut and pursued coding and documentation standard within the group.
- Get a grade of B or better.
    - Minimise our reliance on the supervisor's guidance, and instead independently evaluate its relevance and applicability on a case-by-case basis.
    - Write a well-structured thesis with cogent argumentation, appropriate use of lexicon and meta-literary answers.

Effect goals

- Have a program to be proud of and show in job interviews.
    - Clean and clear cut code
    - Fun/exciting game
    - Good documentation of code

Frame

Time limits:

The time limit for the bachelor thesis is the 18th of May. After this it is still possible to work on the project towards the presentation. We may also work on the program itself after the end of the Bachelor, although if we do we should probably then create one repository of the code each, in case someone in the group does not want there to be any changes.

Our work hours per week will be calculated such that we achieve a total of 300 manhours per person in the group. This is likely to be dynamically adjusted throughout the period in order to meet important milestones.

Scope

Field of study

We will be using our knowledge of program development for organizing the project. Our knowledge of game design for designing the game and our knowledge of programming to implement our game.

Further fields of study we know we will be touching on are:

- Game programming
    - Mathematics
    - Algorithms
    - Artificial Intelligence
        - Predicting player movements and decisions
        - Decision trees
- Game mechanics
    - Game flow
- Game design
    - Balancing
- Professional Programming
    - Documentation

Fields we may touch upon, depending on game design decisions, include:

- History
    - History of war
    - History of weapons and armour
    - Lingual history
- Geography
    - Border history
    - Demographics

- Psychology
    - How wars are started
    - Diplomatic relations and how to manage them

Task description

We want to make a game which appeals to several genres of gaming and the gamers associated with them. To do this, we will need to find the correct mechanics for all genres we mean to touch upon, while not alienating gamers of the other genres. This will therefore require a high level of balancing, as well as deep knowledge within all fields of study as listed in the “Field of study” paragraph.

Limitations

- No online multiplayer, local co-op or split-screen modes
- No centralized server for distributing updates, storing saves or account data
- No Virtual/Augmented Reality headset support for the gameplay
- Possibly not a full/polished game
- Only free art and sound assets
- No spoken dialogue

Project organization

Responsibilities and roles

As we are using a flat structure because of the size of the group we will be using a method for dividing responsibilities and roles. We will be using a method introduced to us by Simon McCallum. Each of us will be given the role of one or more animals and each animal will have its own responsibilities.

- Bear
    - The Leader
    - When no one is speaking, the bear needs to start the conversation.
    - They have the overall vision of the project.
- Wolf
    - The manager
    - Potentially taking notes of who is contributing and ensuring that everyone in the group has a chance to talk.
- Owl
    - The processor
    - Takes notes and watches the schedules.
- Dog
    - The enthusiast
    - Provides solutions to ideas
    - Supports the creation of new ideas.

- Cat
    - The cynic
    - Finds problems to fix to make the project better.
    - Flag problems early on.
- Rabbit
    - The facilitator
    - Watches for problems in the group where people are missing something.
    - Thinks about the infrastructure around the project, and makes sure things are in place to make ideas work.

Bjørn: Bear and Owl.
Bendik: Dog and Cat.
Per Arne: Wolf and Rabbit.

Group Rules

- There is no designated group leader, everyone has an equal say in all matters.
- Signing of documents on behalf of the group should be done with the presence and consent of all members of the group.
- Sanctions against a member who has not fulfilled their quota on work done, will be discussed within the group and adjusted according to the scale of the failed quota.
- For a group member who does not meet expectations continuously expulsion from the group will be considered. For expulsion to happen all the other members have to agree, as well as the supervisor.
- If you can't come to a meeting communicate this in advance.
- If a member of the group disagrees with a decision, the decision will be discussed and voted on. As we are an uneven number of people decision will always be made.
- If a purchase is to be made, it needs to be discussed with the other group members for everyone to split the bill. If the person purchases without asking the group he will have to pay for the expenses themselves.

Planning, monitoring and reporting

Development model

When choosing the development model we were considering the waterfall model as well as extreme programming as development methods. We decided against using the waterfall model as it does not cope well with changes. Using extreme programming as well was not going to work because it requires two and two to be in the same position when writing code.

We decided on the agile development model scrum. This is mostly because it is a development model that we know very well. The daily scrum meeting and weekly sprint meetings also ensure that progress is made, which is essential in the project. As it is an agile development method it is also more capable of handling changes to the project.

The way we will be implementing scrum is that we will have sprints going from wednesday to wednesday. On wednesday we will have the sprint review in person for discussing the sprint and planning the next one. The reason for choosing wednesday was it was the day that was simples for all of us to meet for sprint meetings. Each weekday in the sprint week will have a scrum meeting at 12am each day where we discuss what we are going to do that day. For informing each other about what we got done that day we will be using email messages. This means there will be less time spent in the scrum meetings recapping what we did the day before as we only need to read the emails. The scrum meetings will most likely be done through skype or a similar voip. We do not require members of the group to work in the weekend. They may do so if they like, but no scrum meetings will be held. Each scrum and sprint meeting will be documented in confluence under meetings.

Testing

When development has made such progress that the different components may begin to be tested together. One person in the group will use one day to just debug the various components. This person does not need to be the same every time and we should not test our own code.

We will not be fully committed to doing test driven development. So we do not expect each other to create tests before making the class/function. This is because in often hard to know what test will be needed before actually making the class or function itself. The creator may make test if they think they are needed.

Organisation of quality assurance

Documentation

Meetings and decisions will be documented on confluence.
The bachelor thesis itself will be written in latex.

Documentation of Design:

A game design document will be created and updated as the project is developed.

Code documentation:

All code that we introduce should be documented in a way that it can be converted to doxygen documentation. This means all classes and methods should have doxygen style comments in front of them.

Programming standard:

When developing it is necessary to use a programming standard. By using a programming standard the code becomes more readable, easier to maintain and easier to debug.

While it is possible to create our own programming standard the amount of content in the c++ language to cover is very large. It would therefore be easier to use an existing standard and if need be change the parts where we disagree.

We decided on using the google c++ programming standard as it covers extensively standards in the c++ language.
Link: https://google.github.io/styleguide/cppguide.html

Code documentation syntax:

The documentation of functions and classes will differ from programming standard as we will be using doxygen to generate the documentation for the program. As there are many accepted syntaxes for generating doxygen documentation we have decided to use the following syntax:

/*
@short
*/

We chose /**/ because it was the shortest and simplest to write and @ instead of \ because it stands more out.

Tools

Revision control system:

The repository will be using git as the revision control system, as this was preferred by all members of the group.
For online hosting of the git repository we had to consider using Bitbucket or Github. Both are viable options with not many differences.

We will use Bitbucket as the host site for our repository for this project. We did consider using github and connect it with trello and slack. Unfortunately this would cost 8 dollars a month so we decided to go with bitbucket and connect it with JIRA and confluence instead.

In bitbucket we will create a master repository and from that we will create a dev branch. From the dev branch each user creates their own feature branch. When a feature is done it is merged into the dev branch. the dev branch may only be merged into the master branch when the group agrees.

Project management and Issue tracking:

As we are using agile development in the form of scrum it would be useful to have a project management tool that supports that development method. For this we can chose between a variety of platforms. The ones that we are considering are Trello and JIRA.

We decided to go with JIRA as it integrates with bitbucket and offers some more complex functionality though this makes it slightly harder to use.

We will use JIRA as a scrum board, for issue tracking and sprints. Will (hopefully) be connected to Bitbucket and Confluence.

Documentation:

For creating shared writing document we are using a shared google doc folder. For writing the bachelor thesis itself we decided to use Latex because of it academic standard.

For writing in latex we will either use a Latex editor such as texniccenter or using shareLatex. We will use shareLatex if we are given free access to it.

Confluence was also decided upon because it works together with bitbucket and JIRA. It will also be used for documenting meetings and decisions.

Communication:

We will be using skype as a means of communication between the members of the project. We already have been using skype before so it seemed fine to continue to use it. It will also be used for group meetings if we cannot be gathered in one place. As we are such a small group we do not see any problem with using skype even though it is a private form of communication. Email will also be used for communicating progress between scrums. Other than that we may send messages over the phone if it is needed.

Time management:

All three members of the group are in the Professional Programming course, and will therefore need to track the time spent during the Project.

For this, Toggl and Microsoft Excel will be used as time tracking programs. It is possible to share the time data between members in the group, but we have decided to not use a shared project and keep the information private.

Engine:

Using an engine is recommended when making a game. For this we have two choices: Make our own engine (preferably in C++) or use an existing engine, such as Unreal Engine, Unity, etc. The decision against using our own engine is that it would take too much time away from developing the game to working on the engine.

The decision then lies in what engine to use. There are multiple available, but we were mainly considering Unreal 4 and Unity as they both have reasonable business models. Both have assets available in their online store and are good engines. Although Unity is considered more beginner friendly, in the end we decided to use the Unreal engine because it uses C++ as the coding language. The bulk of the work will likely be done using this tool. We will be using the 4.10 version of the engine.

Compiler:

Visual Studio 2015 will be used in tandem with the Unreal Engine to manage any C++ code that we write for the project and other tools that we use. Mainly because it is found by default by unreal engine and it is the compiler that we have used the most.

Risk Analysis

Identifying and analysing project risk

Priority	Risk	Possibility	Consequences	Risks involved
1	Unavailability of technology	Low	High	
2	Disease	High	Low/ Temporary	
3	Injury	Low	Low/ Medium/ High	
4	Bugs in code	High	Low/ Medium	

Plan for handling risk

1. Change technology to be used, may require some time to do.
2. Wait until group member feels better or try to work while sick
3. Wait until group member feels better or try to work while injured
4. Get help from Professional Programming IMT 3602 class

Plan for implementation

- 01.05.16
    - End of development
    - Focus on writing bachelor thesis
- 18.05.16
    - End of literary bachelor thesis
    - Focus on creating presentation
- 05.16/06.16
    - Presentation deadline (tba)
    - Possible individual polishing of game after this deadline

D Game Design Document

The following pages contain the game design document in PDF format. Some of the features planned early on that were not implemented into the game may still be found there, while features implemented during development may not be mentioned in the document, as it has been mainly used as a note document. Please keep in mind, therefore, that certain aspects of the design of the game are more detailed here than in the thesis, while other aspects are more detailed in the thesis than they are here.

Game Design document

This document was written as a notebook at the beginning of the thesis, but has been rewritten later on to allow for more outside readability.

Engine: ***Unreal Engine 4.10.X***
Name: ***Kingdom: Civil War***

Base description

The game consists of an overworld and a combat world.
The overworld will function like risk, Lord of the Rings: Battle for Middle Earth; War of the Ring, Europa Universalis, or other. The player can spawn units and create buildings in controlled regions.

When the player engages an enemy in battle in a region, they enter the combat world with their units and heroes.

The combat world is a 2d or 3d combat system, where you take soldiers with you from the overworld, in which they are trained. You can use different heroes with abilities, such as a swordfighter, a bowman, a mage, a tank, or even one with a "portal gun" (a.i. assassin 2.0).

Theme

Medieval fantasy (+ apocalyptic region?). Knights versus monsters.

Game modes

Singleplayer

Campaign:

- Conquest with a storyline
- Preset instances
- Unlock more regions, units, heroes and buildings as story progresses
- Preset alliances and wars

Instances/skirmishes:

- Combat in a single region
- All players/AI start at the edges of the map
- Several players/AI possible
- Certain regions allows for more players/AI to join

Conquest/conquer the world:

- All regions unlocked
- All units, buildings and heroes unlocked
- Free-for-all gameplay style, may make and break alliances at will (if both affected parties agree), may declare war at will.

Multiplayer

In multiplayer, there may be too much to consider, such as cheating, safety, server, etc., and it may be dropped. Depends on development time.

Overworld

Description:
The overworld consists of all the regions in the map, and is where most major strategic decisions, such as diplomacy, economics and invasions/defensive operations, take place.

Size:
The size of the overworld is decided by the amount of regions within it. The number of players depends on each map, but is usually determined by the size of the map.

Generation:
Pre-made maps will be made available, with randomly generated maps implemented only if time constraints allow it. At least one map must be made (for the campaign).

Regions

Shape:
Natural regions (natural borders, rivers, mountains, etc.).
We will not use the same shape to create the map each region will have its own shape.

Example map (made in Microsoft Paint):

Colour explanation: Green = Forest/Grassland, Grey = Mountains, White = Snow, Light Green = Farmland, Blue = Water (River/Lake/Ocean), Black = Settlement (Village/Town/City) /Bridge, Beige = Desert, Brown = Region Borders.

Building availability:

To determine the number of available buildings in each region, there are three main options:

1. Preset number, but can be upgraded
    - Number of building slots available depends on development level in the region
    - The player may upgrade the development level by spending a set amount of resources, to allow for more building slots.
    - Allows for more flexible resource management, as well as construction of military based regions that produce units and defend borders and civilian regions that generate resources.
2. Static based on size
    - Set number of building slots available, may not be upgraded. To construct new buildings, existing buildings must be razed.
    - May create specialized regions for resources and military, and also makes certain regions more attractive for players.
3. All building in all regions
    - Everything is available for all players in all regions.
    - Overworld becomes more like a sandbox, shifting focus to the combat world.

We chose option 1.

Buildings, units and heroes

These buildings are suggestions, and which ones to be implemented (if any) may vary depending on changes during development.

- Resource generating building:
    - Food:
        - Farms
        - Butchers
        - Bakery
        - Fishing docks
    - Wood:
        - Woodcutter’s hut
        - Sawmill
    - Stone:
        - Quarry
        - Masonry
    - Iron:
        - Iron mine
        - Smelter
        - Smith
    - Gold:
        - Gold mine
        - Mint
        - Marketplace

        - Traders' guild
- Unit generating building:
    - Barracks:
        - Produces standard melee units such as swordsmen, spearmen and pikemen
    - Archery range:
        - Produces standard ranged units such as bowmen and crossbowmen
    - Stables:
        - Produces standard mounted units such as light and heavy cavalry
        - New units, such as missile cavalry, may be unlocked if specific buildings are also built in the region, such as the archery range
    - Tavern:
        - Produces assassins, rogues, mercenaries (same as regular units, but only cost gold, have higher upkeep and are instantly trained), etc.
        - May also produce a small trickle of gold
    - Mages' guild:
        - Produces mages of all kinds (except heroes)
    - Monsters' cave:
        - Produces monsters
    - Monster's lair
        - Produces high-level monsters
- Stronghold building:
    - Fortress
        - Large defensive structure
        - High defensive properties
        - May be best suited for larger armies
    - Fort
        - Medium defensive structure
        - Some defensive properties
        - Best suited for small armies
    - Tower
        - Small defensive structure
        - Allows for overview of region
        - Best suited for small groups of soldiers
- Hero building, only in capital. If conquered, move it.
    - Palace/ heroes' guild/ castle/ hall of heroes
    - May be combined with regular unit buildings to unlock certain heroes
        - Hero building + barracks = paladin
        - Hero building + archery range = ranger
        - Hero building + stables = knight
        - Hero building + tavern = master assassin
        - Hero building + mages' guild = mage hero
        - Hero building + monsters' cave = demon
        - Hero building + monsters' lair = dragon

Factions

May be divided into humans versus monsters, but only if time allows it, as this would require much more testing and balancing.

Combat world

Things that may be implemented if time allows it, but are not truly needed in the game:

- Multiple floors
    - Makes the world seem more alive
    - May be difficult to implement, as we have yet to gain experience with Unreal Engine 4
- Fog of War
    - Allows for more realism, as it is not normal to see over a hill.
    - Will most likely be quite difficult to implement, may wait with this until the very end.

Decisions made for how combat should work:

Flow:

1. Diablo style
2. RTS (lord of the rings / total war)
3. ~~Turn based (Heroes of might and magic)~~
4. ~~Turn based (Worms)~~
5. ~~Turn based (1066)~~
6. Turn based (XCOM)
7. Turn based (age of wonders)
8. Turn based (Endless legend)
9. Hybrid style (Divinity: Original Sin, Valkyria Chronicles)
10. ~~2D hack and slash~~
11. Mount and blade

Unit control:

1. Control hero only (MOBA, Dynasty Warriors)
2. Control whole army (TW, BfM, HMM, MoW, XCOM)
3. Control hero only, but place orders at start of match (Spartan)
4. Control heor only, but give “vague” orders to troops (Reality)

When battles start, you zoom in on the region where the battle will take place, and turn that region into the battle ground.

Flow within this restriction:

1. Turn based (Endless Legend)
2. Hybrid style (Divinity: Original Sin, Valkyria Chronicles)
3. Turn based (XCOM)
4. RTS (lord of the rings / total war)

5. Diablo style
6. Turn based (age of wonders)
7. Mount and blade

Auto combat

Needed in the case of multiplayer or if the player has an overwhelming advantage and does not want to play the match manually.

~~Other option:~~
~~Make the combat like xcom and make the overworld be a more standard party moving around completing quests and slaying monsters.~~

Base Unit Design

1 Iteration of design:
Each unit has attacks added to them at the start of the turn. Each attack is a base attack of iteration 1. The player places all the attacks into an array and has an int to the current selected attack. By default the first attack added is the one that the player starts with.
On selecting new attacks we check if the new attack is different. We then hide the old attack mesh and show the new one, then set the Current attack. The unit can also attack another character and take damage. Each unit has a base attack variable that they apply to the damage of the attack and also have a defence variable to reduce damage taken. The unit also contains a decal for knowing showing that it is selected.

2 Iteration of design:
Currently works mostly the same as before but now stats such as defence and health is placed into a collection of stas. Each stat is placed into a stat class or one of its subclasses. Depending on the class the stat may be given modifiers, scaled with level, be a derived stat or be a vital stat. The unit no longer uses attacks it now just has two pointers to two weapons. On is the main hand and the second is the off hand. The weapons are added and equiped based on what is selected in the editor. Now instead the weapon applies its damage when attacking, for more see base attack iteration 2. Attacks have been changed to skills. Also worth mentioning is that some stats such as resists are percentages and stats like sight and movement are in cm.

Stat types:

Type	Description
Stat	Base type. Only stores a float value that can be set to a value.
Modifiable Stat	Contains a list of modifiers that can be

	applied to the stat. This generates a new value that is added when retrieving the stat value. Parent: Stat
Scalable Stat	Contains a float value that can be set to increase the final value of the stat. Such as when we level up we increase the scalable value. Parent: Modifiable Stat
Linkable Stat	Contains a list of other stats that it get a linked value from. So when getting the value of the stat the linked value is added Parent: Modifiable Stat
Vital Stat	Contains a current value that is within the range of the final value and 0. Parent: Linkable Stat

All current stats that a unit has:

Stat name	Stat type
Strength	Scalable Stat
Dexterity	Scalable Stat
Intelligence	Scalable Stat
Constitution	Scalable Stat
Speed	Scalable Stat
Perception	Scalable Stat
Movement	Linkable Stat
Sight	Linkable Stat
Health	Vital Stat
ActionPoints	Vital Stat
FireResistence	Modifiable Stat
EarthResistance	Modifiable Stat
WaterResistance	Modifiable Stat
AirResistance	Modifiable Stat

ChriticalHitChance	Linkable Stat
ChriticalHitMultiplier	Linkable Stat

Linkable and vital stats are derived from

Stat	Derived from
Movement	(Speed * 0.5)
Sight	(Perception * 2.0)
ChriticalHitChance	(Perception * ratio) ?
ChriticalHitMultiplier	Dex?
Health	Constitution * 5
ActionPoints	Constitution * 1
ActionPoints Regen	(Speed * 0.5)

The code structure for this is like the following class diagram
(Insert class diagram)

Base Attack Design

1 Iteration of design:
Each attack is a subclass of the base attack class. Each attack is has its own Action point cost, range, damage, attack type, socket name for where to connect to the skeleton and an attack icon texture for the attack hotbar. Each subclass sets their own default values. There are currently no functions for the attack class.

2 Iteration of design:
Instead of having attack be both a weapon and a skill i separated the two into weapons and skills. This is now the design for the weapons. Much remains the same, but socket name has been moved from the weapon to the base character instead. There is also no longer an attack icon. The damage has changed to minimum and maximum damage. I have added damage type, weapon type and a boolean for if the weapon is two handed. Another major new feature is that weapons also now contain an array of stats that they apply to the character, example: +5 to dexterity or +10% to health. These are applied to the character as he equips the weapon and removed when unequipped.

3 Iteration of design:

Each weapon has an attack type to know if it is a melee, ranged or magic weapon. Each weapon also contains an attribute that is used to add to the hit chance.

Example: The base hit chance is 40% then we check what attribute the weapon has say strength. We then take the character's strength * a ration and add it to the total hit chance. Say you have 10 strength and the ratio is 6% then we add 60% to the hit chance making it 100%.

The final hit chance is then reduced by the enemies dodge chance.
Another possibility was to add hit chance to the weapons instead.Finally the weapon attribute type also affects damage by taking the attribute value * a ratio and adding it to the total damage.

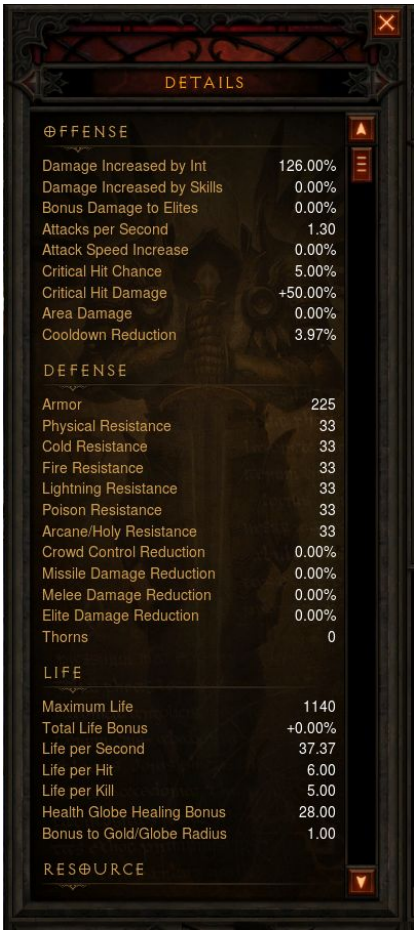
Base Skill Design

Unit stat menu design:

1 Iteration of design:
Placed stats in a vertical box starting with the unit name. Then in order bellow added health, action points, Base attack and base defence.

2 Iteration of design:

With the changes to the characters how we display the stats needed to change. First Health and action points is now moved over to the action bar. The first thought was to do the stats similar to Divinity original sin, but we changed instead into a diablo 3 styled stat menu.



Other considerations are if the unit (m, m/s) should be its own widget in the menu therefor constant or if it should be appended to the value when retrieved. This is mainly a memory or processing question. Currently as there is no need to optimize the unit or any other constant values will be set in the editor rather than appended in code.

The stat Menu is a little big so it will only be visible by clicking a button on the action bar to display it. The same applies to health bars, they will also only appear when you select a unit.

E Decisions and Notes

The following pages contain the decisions and notes document that has been used for common notes and decision-making for our project in PDF format. This document is note-heavy, as it was used to write down ideas, several of which were rejected, and is not a professional document in any way. Instead, it should be only be viewed as a way to see into the minds of the authors during early development.

Decisions and notes

List of desireable genres:

- Action
    - ~~Platform~~
    - ~~Shooter~~
        - ~~Shoot'em up~~
        - ~~Deathmatch / Team Deathmatch~~
    - Beat'em up/Hack and slash

- ~~Action-Adventure~~
    - ~~Metroidvania~~
    - ~~Platform~~

- Adventure
    - Real-time 3D adventure

- Role-playing
    - Action RPG
    - Rogue like
    - Tactical RPG
    - SandBox RPG
    - ~~(WRPG/JRPG)~~

- Strategy
    - 4X
    - ~~Real-time strategy~~
    - ~~Real-time tactics~~
    - Turn based strategy
    - Turn based tactics
    - Wargame
    - ~~Puzzle~~

Game ideas:

- ~~Metroidvania style game using world builder tool (AR/VR) to move player/stop player/hinder player/aid player.~~
    - ~~Co-op?~~

- Strategic overworld/campaign map (turn-based), with RPG real-time events (Total War/ Might and Magic combined with an RPG)
    - Campaign map made/remade using world builder tool?

- ~~Remake of game programming project, OutOfMemory (worms clone) in Unreal~~

- Tactics game using world builder tool (AR/VR) to create the battlefield then apply a grid (Square or Hexagon) for units to move in. Or possibly just have it be real time.

- ~~Minecraft mod?~~

- 4x game focusing on exploration and extermination.
- Action rpg, think diablo and such
- ~~Isometric rpg, think baldur's gate and pillars of eternity~~

- Bacteria rts, player plays as the immune system defending the body.

- Action-Adventure / Puzzle game with dynamic (destructible or changing) world, possibly non-euclidean or otherwise control the perception of the world

- Portal Worms

Phase 2:

- “Lord of the rings: battle for middle earth; war of the ring” overworld and some either 2d or 3d combat system, where you take soldiers with you from the overworld, in which they are trained. Using different heroes with abilities, such as one with a "portal gun".
- ~~Portal Worms~~
- ~~Tactics game using world builder tool (AR/VR) to create the battlefield then apply a grid (Square or Hexagon) for units to move in. Or possibly just have it be real time.~~
- ~~Action-Adventure / Puzzle game with dynamic (destructible or changing) world, possibly non-euclidean or otherwise control the perception of the world.~~
- ~~4x game focusing on exploration and extermination.~~
- ~~Action rpg, think diablo and such.~~
- ~~Bacteria rts, player plays as the immune system defending the body.~~

Final:

- “Lord of the rings: battle for middle earth; war of the ring” overworld and some either 2d or 3d combat system, where you take soldiers with you from the overworld, in which they are trained. Using different heroes with abilities, such as one with a "portal gun".

Other decisions:

- Virtual support
    - No virtual
        - More focus on game
        - Easier to implement
    - ~~Oculus Rift~~
        - ~~More in-depth~~
    - ~~Virtual environment/ world builder~~
        - ~~Added mechanics~~
        - ~~Innovative~~

- ~~Special controls~~
    - ~~Mobile~~
    - ~~Kinect~~
    - ~~Move~~
    - ~~PS controller~~
    - ~~Wii-remote~~
    - ~~Voice~~
- Network
    - Single-player
    - Co-op
    - Multi-player
        - LAN
        - Massively multiplayer
    - ~~Laptops/Mobile phones~~
- World space
    - ~~2D~~
        - ~~Top-down~~
        - ~~Isometric~~
        - ~~Sidescroller~~
    - 3D
    - ~~Non-euclidean (Portal / Antichamber)~~
    - ~~Hybrid (FEZ)~~

- Game Engine
    - Unreal engine
        - Faster
        - Easier to explain “Why?”
        - Can go in-depth as necessary, rather than recreate whole engine
    - ~~Own engine~~
        - ~~“More to brag about”~~
        - ~~More specialized~~
        - ~~Potential for learning more~~

- Repo
    - Bitbucket
        - More familiar
    - ~~GitHub~~
        - ~~Can connect to Slack~~

- Sprint board
    - Jira
        - Easier to log time spent
        - Easier to log issues
    - ~~Trello~~
        - ~~Easier to keep track of issues~~
        - ~~Connectivity to Slack~~

- Report writing tool
    - Latex
        - Easy to refer to code
        - Type:
            - Shared
            - ~~In repo~~
                - ~~Rules to avoid merge conflict~~
    - ~~Confluence~~
        - ~~Easy to refer to issues~~
- Time usage recording:
    - Toggl.com
    - Excel

Engine Choice

Decided to use unreal.
Using version 4.10.1 as it's the latest that is not a preview.

Latex:

Decision between using Latex shared and having latex in the repo.
~~Plus shared latex is not payed for by the school.~~
~~We decided to have latex in the repo.~~

~~**To install latex editor:**~~
~~First download MikTex for converting .tex to pdf. Simplest to install it for all users else you need to remember where the bin folder is.~~
~~Link:~~ http://www.miktex.org/download

~~Download and install sumatra pdf as it is faster than adobe for previewing the generated pdf.~~
~~Link:~~ http://www.sumatrapdfreader.org/free-pdf-reader.html

~~The download TexnicCenter that is the editor. After installed perform the config. If the MikTex is installed in the c folder TexnicCenter finds the bin automatically and finds sumatra automatically else you will have to find it for it.~~
~~Link: http://www.texniccenter.org/download/~~

ShareLatex now payed for by school. Easy to use, no repo required, easy for supervisor to read and comment on work. https://www.sharelatex.com/

Coding style:

For visual studio make sure to use same number of spaces in tabs, 4. Decided by popularity.

For documentation use the following syntax:
/*
@
@
*/

Other syntaxes was considered such as:
///
or
/**
*
*/
These were decided against as /// on every line was harder than doing /**/ and the other was decided against because it was java syntax.
We are also using @ instead of \ because it is cleaner.

Brackets are placed after the function/classes the following way
foo()
{

}

not
foo(){

}
as this is java convention.

Also data members in classes should be declared the following way:
class Foo
{
int mName;
};

with a lowercase m to indicate that it is a member of the object.

Repo method:

One main master repo.
One (Forked/Branched?) Dev repo.
One (Forked/Branched?) version of the dev repo for each user to implement new functionality.

Once the functionality is implemented the (Forked/Branched?) is merged back into the dev branch and the user will need to create a new (Forked/Branched?) for implementing new functionality.

Other suggestions from mariusz and simon:
One person is responsible for pulling dev into master. Anyone may push into dev, but pull requests from dev to master is made as a group.

Development method:

Why we chose scrum?

Scrum:

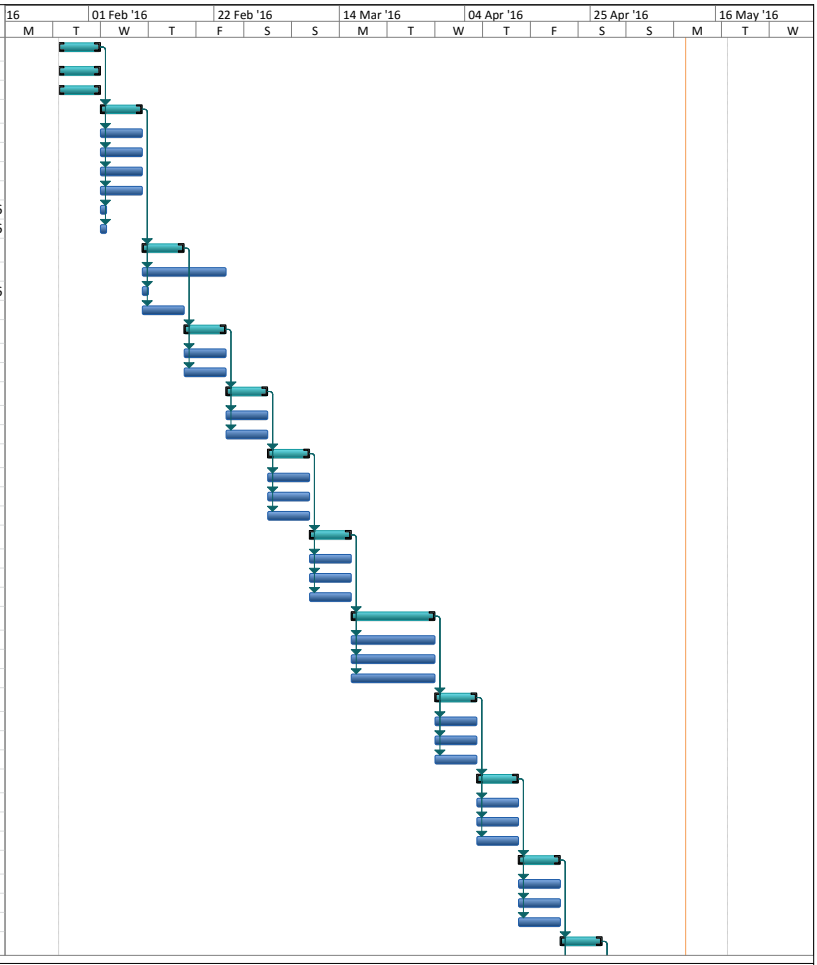
Sprint duration 1 week each. Going from wednesday to Wednesday.
Sprint review in person each Wednesday.
~~Scrum at the start of each day at 12 either through skype or other means of communication. Instead of reporting on the progress from the day before use some messaging system like email or commits?~~
No scrums in the weekend.
Write down a summary of each meeting in written format.

Testing:
Unreal has inbuilt testing, apparently.
Probably only need testing on individual classes, not interactions between.
Have one person doing a days of testing on the interactions between the modules.

F Gantt Charts

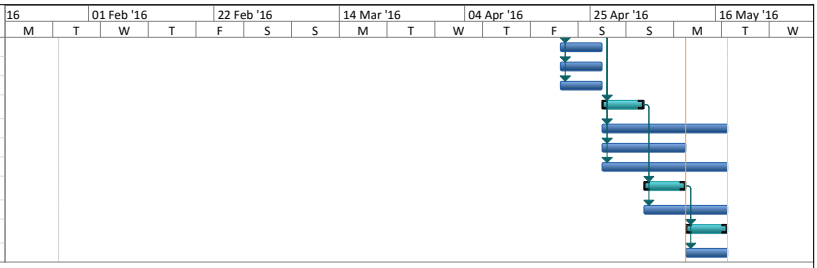
The following pages contain the two Gantt charts relevant to this project. The first Gantt chart shows our initial plan of implementation, and ideally would have been the chart to follow. However, because of the large amount of changes and restructuring, the final implementation changed drastically, and the second Gantt chart therefore shows how the game was actually implemented.

ID	Task Name	Duration	Start	Finish
1	***Initial Sprint***	5 days	Wed 27.01.16	Tue 02.02.16
2	Write and hand in project contract	5 days	Wed 27.01.16	Tue 02.02.16
3	Write and hand in front project	5 days	Wed 27.01.16	Tue 02.02.16
4	***Initializer Sprint***	5 days	Wed 03.02.16	Tue 09.02.16
5	Setup Latex document	5 days?	Wed 03.02.16	Tue 09.02.16
6	Setup web page	5 days?	Wed 03.02.16	Tue 09.02.16
7	Work on Game Design document	5 days?	Wed 03.02.16	Tue 09.02.16
8	Learn basic Latex scripting	5 days	Wed 03.02.16	Tue 09.02.16
9	Draw basic map	1 day	Wed 03.02.16	Wed 03.02.16
10	Contact extrenal (Sarepta)	1 day?	Wed 03.02.16	Wed 03.02.16
11	***Sprint 1***	5 days	Wed 10.02.16	Tue 16.02.16
12	Unreal Engine 4 research and training	10 days	Wed 10.02.16	Tue 23.02.16
13	Bitbucket repo and branch setup	1 day	Wed 10.02.16	Wed 10.02.16
14	Unreal Engine asset research and importing	5 days	Wed 10.02.16	Tue 16.02.16
15	***Sprint 2***	5 days	Wed 17.02.16	Tue 23.02.16
16	Menu setup	5 days	Wed 17.02.16	Tue 23.02.16
17	Testing level setup	5 days	Wed 17.02.16	Tue 23.02.16
18	***Sprint 3***	5 days	Wed 24.02.16	Tue 01.03.16
19	Actor/Character prototypes	5 days	Wed 24.02.16	Tue 01.03.16
20	Advanced test levels	5 days	Wed 24.02.16	Tue 01.03.16
21	***Sprint 4***	5 days	Wed 02.03.16	Tue 08.03.16
22	Basic AI	5 days	Wed 02.03.16	Tue 08.03.16
23	Actor/Character offensive capabilities	5 days	Wed 02.03.16	Tue 08.03.16
24	Basic terrain	5 days	Wed 02.03.16	Tue 08.03.16
25	***Sprint 5***	5 days	Wed 09.03.16	Tue 15.03.16
26	Multiple actors/characters	5 days	Wed 09.03.16	Tue 15.03.16
27	AI offensive capabilities	5 days	Wed 09.03.16	Tue 15.03.16
28	Grass, trees, basic graphical textures	5 days	Wed 09.03.16	Tue 15.03.16
29	***Sprint 6 (Week + Easter)***	10 days	Wed 16.03.16	Tue 29.03.16
30	Items	10 days	Wed 16.03.16	Tue 29.03.16
31	AI multiple enemy actor tracking	10 days	Wed 16.03.16	Tue 29.03.16
32	Level streaming	10 days	Wed 16.03.16	Tue 29.03.16
33	***Sprint 8***	5 days	Wed 30.03.16	Tue 05.04.16
34	Stats (armour, resistance, etc.)	5 days	Wed 30.03.16	Tue 05.04.16
35	AI Cooporation	5 days	Wed 30.03.16	Tue 05.04.16
36	Larger levels	5 days	Wed 30.03.16	Tue 05.04.16
37	***Sprint 9***	5 days	Wed 06.04.16	Tue 12.04.16
38	Special attacks	5 days	Wed 06.04.16	Tue 12.04.16
39	AI stats and item integration	5 days	Wed 06.04.16	Tue 12.04.16
40	Battle start in regions, auto zoom to combat area	5 days	Wed 06.04.16	Tue 12.04.16
41	***Sprint 10***	5 days	Wed 13.04.16	Tue 19.04.16
42	Conquest + skirmish menu finilization	5 days	Wed 13.04.16	Tue 19.04.16
43	AI special attacks integration	5 days	Wed 13.04.16	Tue 19.04.16
44	Buildings integration	5 days	Wed 13.04.16	Tue 19.04.16
45	***Sprint 11***	5 days	Wed 20.04.16	Tue 26.04.16



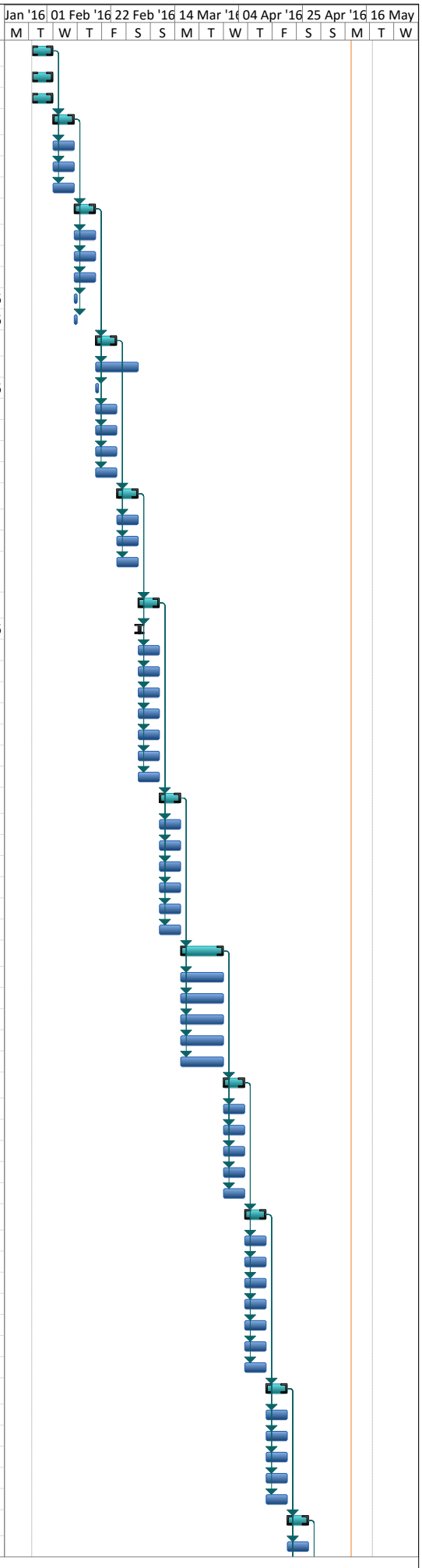
Project: Gantt chart

ID	Task Name	Duration	Start	Finish
46	Campaign menu finilization, graphics menu, keybindings	5 days	Wed 20.04.16	Tue 26.04.16
47	Overworld AI	5 days	Wed 20.04.16	Tue 26.04.16
48	Graphics polishing	5 days	Wed 20.04.16	Tue 26.04.16
49	***Sprint 12***	5 days	Wed 27.04.16	Tue 03.05.16
50	Bug fixing	15 days	Wed 27.04.16	Tue 17.05.16
51	Testing	10 days	Wed 27.04.16	Tue 10.05.16
52	Bachelor report	15 days	Wed 27.04.16	Tue 17.05.16
53	***Sprint 13***	5 days	Wed 04.05.16	Tue 10.05.16
54	Bachelor report fixes	10 days	Wed 04.05.16	Tue 17.05.16
55	***Sprint 14***	5 days	Wed 11.05.16	Tue 17.05.16
56	Bachelor report polishing	5 days	Wed 11.05.16	Tue 17.05.16



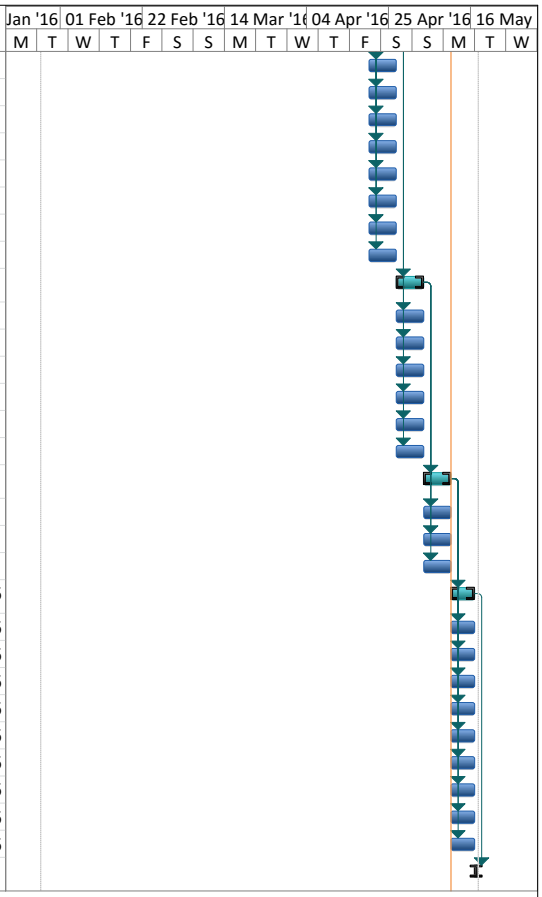
Project: Gantt chart

ID	Task Name	Duration	Start	Finish
1	***Initial Sprint***	5 days	Wed 27.01.16	Tue 02.02.16
2	Write and hand in project contract	5 days	Wed 27.01.16	Tue 02.02.16
3	Write and hand in front project	5 days	Wed 27.01.16	Tue 02.02.16
4	***Initializer Sprint***	5 days	Wed 03.02.16	Tue 09.02.16
5	Setup webpage	5 days	Wed 03.02.16	Tue 09.02.16
6	Work on game design document	5 days	Wed 03.02.16	Tue 09.02.16
7	Setup Latex thesis document	5 days	Wed 03.02.16	Tue 09.02.16
8	***Setup Sprint***	5 days	Wed 10.02.16	Tue 16.02.16
9	Learn basic Latex scripting	5 days	Wed 10.02.16	Tue 16.02.16
10	Main Menu branch setup	5 days	Wed 10.02.16	Tue 16.02.16
11	Terrain setup research	5 days	Wed 10.02.16	Tue 16.02.16
12	Draw basic map	1 day	Wed 10.02.16	Wed 10.02.16
13	Contact extrenal (Sarepta)	1 day	Wed 10.02.16	Wed 10.02.16
14	***A New Hope***	5 days	Wed 17.02.16	Tue 23.02.16
15	Unreal Engine 4 research and training	10 days	Wed 17.02.16	Tue 01.03.16
16	Bitbucket repo and branch setup	1 day	Wed 17.02.16	Wed 17.02.16
17	Unreal Engine asset research and importing	5 days	Wed 17.02.16	Tue 23.02.16
18	Asset finding and importing	5 days	Wed 17.02.16	Tue 23.02.16
19	Settings and Campaign menu layout	5 days	Wed 17.02.16	Tue 23.02.16
20	Create test map	5 days	Wed 17.02.16	Tue 23.02.16
21	***The Empire Strikes Back***	5 days	Wed 24.02.16	Tue 01.03.16
22	Make tile prototype	5 days	Wed 24.02.16	Tue 01.03.16
23	Research AI	5 days	Wed 24.02.16	Tue 01.03.16
24	Make temporary overworld map for streaming tests	5 days	Wed 24.02.16	Tue 01.03.16
25	***Return of the Jedi***	5 days	Wed 02.03.16	Tue 08.03.16
26	*Sarepta meeting*	1 day	Wed 02.03.16	Wed 02.03.16
27	Research advanced AI	5 days	Wed 02.03.16	Tue 08.03.16
28	Fill in content on web page	5 days	Wed 02.03.16	Tue 08.03.16
29	Implement sound options	5 days	Wed 02.03.16	Tue 08.03.16
30	Create Campaign UI	5 days	Wed 02.03.16	Tue 08.03.16
31	Create a unit that can be moved	5 days	Wed 02.03.16	Tue 08.03.16
32	Create landscape for test Campaign	5 days	Wed 02.03.16	Tue 08.03.16
33	Create sub-level for a region	5 days	Wed 02.03.16	Tue 08.03.16
34	***The Phantom Menace***	5 days	Wed 09.03.16	Tue 15.03.16
35	Integrate level streaming into Campaign	5 days	Wed 09.03.16	Tue 15.03.16
36	Add melee attack to units	5 days	Wed 09.03.16	Tue 15.03.16
37	Work on terrain	5 days	Wed 09.03.16	Tue 15.03.16
38	Improve base of unit character	5 days	Wed 09.03.16	Tue 15.03.16
39	Switch turns between teams	5 days	Wed 09.03.16	Tue 15.03.16
40	Implement basic AI	5 days	Wed 09.03.16	Tue 15.03.16
41	***Attack of the Clones***	10 days	Wed 16.03.16	Tue 29.03.16
42	Create region borders	10 days	Wed 16.03.16	Tue 29.03.16
43	Continue on terrain	10 days	Wed 16.03.16	Tue 29.03.16
44	Improve UI for units	10 days	Wed 16.03.16	Tue 29.03.16
45	Redesign units and their attacks	10 days	Wed 16.03.16	Tue 29.03.16
46	Make AI track allies	10 days	Wed 16.03.16	Tue 29.03.16
47	***Revenge of the Sith***	5 days	Wed 30.03.16	Tue 05.04.16
48	Imlement action point cost	5 days	Wed 30.03.16	Tue 05.04.16
49	AI specified enemy targeting	5 days	Wed 30.03.16	Tue 05.04.16
50	Add foliage	5 days	Wed 30.03.16	Tue 05.04.16
51	Calculate hit chance	5 days	Wed 30.03.16	Tue 05.04.16
52	Display combat hit chance	5 days	Wed 30.03.16	Tue 05.04.16
53	***The Force Awakens***	5 days	Wed 06.04.16	Tue 12.04.16
54	Create a minimap	5 days	Wed 06.04.16	Tue 12.04.16
55	AI item/ special attack framework	5 days	Wed 06.04.16	Tue 12.04.16
56	Create unit spawners	5 days	Wed 06.04.16	Tue 12.04.16
57	Attempt to fix AI bugs	5 days	Wed 06.04.16	Tue 12.04.16
58	Create 10 melee units and weapons	5 days	Wed 06.04.16	Tue 12.04.16
59	Improve skirmish controller	5 days	Wed 06.04.16	Tue 12.04.16
60	Fix select on player	5 days	Wed 06.04.16	Tue 12.04.16
61	***Knights of the Old Repuplic***	5 days	Wed 13.04.16	Tue 19.04.16
62	Add victory condition	5 days	Wed 13.04.16	Tue 19.04.16
63	Add animations to humanoid units	5 days	Wed 13.04.16	Tue 19.04.16
64	Add loading screen	5 days	Wed 13.04.16	Tue 19.04.16
65	Create a demo	5 days	Wed 13.04.16	Tue 19.04.16
66	Add damage types	5 days	Wed 13.04.16	Tue 19.04.16
67	***Republic Commando***	5 days	Wed 20.04.16	Tue 26.04.16
68	Instance/skirmish menu layout	5 days	Wed 20.04.16	Tue 26.04.16



Project: Gantt_final_chart

ID	Task Name	Duration	Start	Finish
69	Add ranged attack	5 days	Wed 20.04.16	Tue 26.04.16
70	Do a performance analysis	5 days	Wed 20.04.16	Tue 26.04.16
71	Add combat log	5 days	Wed 20.04.16	Tue 26.04.16
72	Add snow, mountains and farmland	5 days	Wed 20.04.16	Tue 26.04.16
73	Add village/-s	5 days	Wed 20.04.16	Tue 26.04.16
74	Push Campaign map	5 days	Wed 20.04.16	Tue 26.04.16
75	Make units turn towards the unit they are attacking	5 days	Wed 20.04.16	Tue 26.04.16
76	Make units unable to attack trough walls	5 days	Wed 20.04.16	Tue 26.04.16
77	***Empire at War***	5 days	Wed 27.04.16	Tue 03.05.16
78	AI integrate attack	5 days	Wed 27.04.16	Tue 03.05.16
79	Add AI damage types	5 days	Wed 27.04.16	Tue 03.05.16
80	Improve game based on feedback	5 days	Wed 27.04.16	Tue 03.05.16
81	Reduce repository size	5 days	Wed 27.04.16	Tue 03.05.16
82	Resolve lighting issues	5 days	Wed 27.04.16	Tue 03.05.16
83	Start work on thesis	5 days	Wed 27.04.16	Tue 03.05.16
84	***Rogue One***	5 days	Wed 04.05.16	Tue 10.05.16
85	Implement further improvements	5 days	Wed 04.05.16	Tue 10.05.16
86	Balance game further	5 days	Wed 04.05.16	Tue 10.05.16
87	Continue work on thesis	5 days	Wed 04.05.16	Tue 10.05.16
88	***The Saga Ends***	4 days	Wed 11.05.16	Mon 16.05.16
89	Finish up and merge final cut of AI	4 days	Wed 11.05.16	Mon 16.05.16
90	Finish up final cut of Battle selection	4 days	Wed 11.05.16	Mon 16.05.16
91	Final improvements	4 days	Wed 11.05.16	Mon 16.05.16
92	Clean up code	4 days	Wed 11.05.16	Mon 16.05.16
93	Clean up design documents	4 days	Wed 11.05.16	Mon 16.05.16
94	Make tutorial	4 days	Wed 11.05.16	Mon 16.05.16
95	Make gameplay video	4 days	Wed 11.05.16	Mon 16.05.16
96	Game testing with classmates	4 days	Wed 11.05.16	Mon 16.05.16
97	Finish thesis	4 days	Wed 11.05.16	Mon 16.05.16
98	***Hand in thesis***	1 day	Tue 17.05.16	Tue 17.05.16



G Meeting Logs

The following pages contain all meeting logs written on Confluence printed out in PDF. These meetings were (mostly) held on Wednesdays from 12:00, with varying degrees of length.

Meeting Notes

Add Meeting Notes

Title	Creator	Modified
2016-05-11 Sprint Meeting Notes	Bjørn Nødland Fuglestad	May 11, 2016
2016-05-04 Sprint Meeting Notes	Bjørn Nødland Fuglestad	May 04, 2016
2016-04-27 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Apr 30, 2016
2016-04-20 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Apr 20, 2016
2016-04-13 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Apr 13, 2016
2016-04-06 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Apr 06, 2016
2016-03-30 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Mar 30, 2016
2016-03-16 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Mar 16, 2016
2016-03-09 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Mar 11, 2016
2016-03-02 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Mar 02, 2016
2016-03-02 Sarapta Meeting Notes	Bjørn Nødland Fuglestad	Mar 02, 2016
2016-02-24 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Feb 24, 2016
2016-02-17 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Feb 17, 2016
2016-02-10 Sprint Meeting Notes	Bjørn Nødland Fuglestad	Feb 10, 2016
2016-02-05 Meeting Notes	Bjørn Nødland Fuglestad	Feb 10, 2016
2016-02-03 Meeting Notes	Bjørn Nødland Fuglestad	Feb 03, 2016
2016-01-27 Meeting Notes	Bjørn Nødland Fuglestad	Jan 27, 2016
2016-01-13 Meeting Notes	Bjørn Nødland Fuglestad	Jan 15, 2016
2016-01-11 Meeting Notes	Bjørn Nødland Fuglestad	Jan 15, 2016
2016-01-07 Meeting Notes	Bjørn Nødland Fuglestad	Jan 15, 2016

Find more results

2016-01-05 Meeting Notes

Date

5 January 2016

Attendees

- Bjørn Nødland Fuglestad
- Per-Arne Waaler Stenshagen

Goals

- Come closer to deciding on what the game for the thesis will be about.

Discussion Items

Time	Item	Who	Notes
45min	Game genres	All	Decided against simulator, sports and management game, but as Bendik Hillestad is not here not here no category is removed.

2016-01-06 Meeting Notes

Date

6 January 2016

Attendees

- Bjørn Nødland Fuglestad
- Per-Arne Waaler Stenshagen
- Bendik Hillestad

Goals

- Eliminate genres that none of us want to do.

Discussion Items

Time	Item	Who	Notes
27min 52s	Eliminate genres	All	• Removed Simulation • Removed Sport • Removed Management game

2016-01-07 Meeting Notes

Date

7 January 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen
- Simon McCallum

Goals

- Come closer to a game idea.
- Discuss tools to use in the project

Discussion Items

Time	Item	Who	Notes

1 hour	Removed sub-genres	Bjørn Nødland Fuglestad Bendik Hillestad Per-Arne Waaler Stenshagen	• Removed sub-genres in the remaining categories

1 hour	Discussed tool to use	Bjørn Nødland Fuglestad
		Bendik Hillestad
		Per-Arne Waaler Stenshagen
			• Repo tool
			  • Github
			  • Bitbucket
			• Game engine
			  • Our own
			  • Unreal
			  • Unity
			• Sprint board
			  • JIRA
			  • Trello
			• Documentation tool
			  • Latex
			  • Confluence

45 min	What the game should be	Simon McCallum	• World
			  • 3D
			  • 2D
			• Virtual support
			  • Oculus
			  • World builder
			• Special controls

2016-01-11 Meeting Notes

Date

11 January 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen
- Simon McCallum
- Mariusz Nowostawski

Goals

- Discuss development method
- Continue discussing tools

Discussion Items

Time	Item	Who	Notes

30 min	Development method	• Bjørn Nødland Fuglestad
		• Bendik Hillestad
		• Per-Arne Waaler Stenshagen
			• Waterfall
			• Extreme programming
			
			• Scrum
			  • Sprints Wednesday to Wednesday
			  • Nothing in the weekend
			  • Daily scrums, meetings at 12:00

15 min	Repo methodology	• Bjørn Nødland Fuglestad
		• Bendik Hillestad
		• Per-Arne Waaler Stenshagen
			• Master repo
			  • Dev branch
			  • feature branch

30 min	Tools	• Bjørn Nødland Fuglestad • Bendik Hillestad • Per-Arne Waaler Stenshagen	• Time management

45 min	Discussion with supervisors.	• Bjørn Nødland Fuglestad
		• Bendik Hillestad
		• Simon McCallum
		• Mariusz Nowostawski
			• Scrum
			  • Messaging system for informing participants on progress.
			• Testing
			  • Unreal inbuilt testing
			• Toggl.com for time tracking

2016-01-13 Meeting Notes

Date

13 January 2016

Attendees

- Bjørn Nødland Fuglestad
- Per-Arne Waaler Stenshagen
- Bendik Hillestad

Goals

- Decide on the game

Discussion Items

Time	Item	Who	Notes
1 hour 30 min	Tools	All	• No longer slack and trello • Going to use JIRA and confluence
1 hours	Game	All	• Decided on the general idea of the game
1 hour	Game design document	All	• More decisions need to be made before it is done.

2016-01-27 Meeting Notes

Date

27 January 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish Project plan
- Deliver the Project agreement

Discussion Items

Time	Item	Who	Notes
45m	Project agreement	All	• Finished printing and signing

2h 40min	Project plan	All	• Almost done
			• Need
			  • Group rules
			  • Scope
			  • Programming standard
			  • Testing
			  • Ducumentation

2016-02-03 Meeting Notes

Date

03 February 2016

Attendees

- Bjørn Nødland Fuglestad
- Per-Arne Waaler Stenshagen

Goals

- Discuss game design
- Start first sprint.
- Integrate Bitbucket, JIRA and Confluence.
- Discuss external meeting.

Discussion Items

Time	Item	Who	Notes
15 min	What to do going forward	Bjørn Nødland Fuglestad Per-Arne Waaler Stenshagen	• ShareLatex
1 hour	Creating sprint for next week	Bjørn Nødland Fuglestad Per-Arne Waaler Stenshagen	• JIRA

2016-02-05 Meeting Notes

Date

05 February 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Notes

- Talked about how to structure the latex thesis document.
  - Will use the subfile package for now.
- Talked about logging scrums, sprints and work notes in google docs.
  - Talked with mariusz as well.
  - Logging scrums and sprints are good, work notes are probably overkill.

- Talked game design.

Discussion Items

Time	Item	Who	Notes
2 hour 5 min	Intermediate discussion	Bjørn Nødland Fuglestad Bendik Hillestad Per-Arne Waaler Stenshagen	SharedLatex Jira

2016-02-10 Sprint Meeting Notes

Date

10 February 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Notes

- **Design**
    - Map
        - Zoom in for combat
    - Combat
        - Units and heroes
        - Not decided if tiles or not
    - Edited the game design document.
- Sprint
    - Finish old sprint
    - New implementation
        - Main menu
        - Settings menu
    - Design some maps
    - Find assets
    - Research terrain generation
- Sarapta
    - Still no respons, might send new mail at end of week

Discussion Items

Time	Item	Who
1 hour	Discussing game design	All
35 min	Discussing game design	Mariusz
1 hour	Discussing game design	All
30 min	Setup next sprint	All

2016-02-17 Sprint Meeting Notes

Date

17 February 2016

Attendees

- Bjørn Nødland Fuglestad
- Per-Arne Waaler Stenshagen
- Bendik Hillestad

Notes

- Blueprints
    - Use when it supported.
- Fished the setup sprint.
- Start next sprint A New Hope

Discussion Items

Time	Item	Who
1h 30min	Discussing what we did this sprint. Finishing the report for the old sprint.	All
22 min	What we have done and what to do	Mariusz
30	Setup of next sprint	All

2016-02-24 Sprint Meeting Notes

Date

24 February 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Notes

Did:

- Assets
- Dynamic loading of levels
- Discuss moving information between levels in unreal.
- Implementing menus

Will do:

- Create tile prototype
- Learn about AI in unreal
- Create a temporary over world map.
- Create a campaign level.

Discussion Items

1h	Finish old sprint	All
1h	Create new sprint	All

2016-03-02 Sarapta Meeting Notes

Date

02 March 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen
- Catharina Lupo

Goals

- Discuss the game
- Get information about further development

Discussion Items

Time	Item	Who	Notes
20 min	Introducing the game	All	• Map • Combat • Menus • Ideas

20 min	Discussing the game	All	• Least viable product
			• Dont bite over to much
			• Creating a company
			• How to comunicate further
			  ◦ Write sprint summarys
			  ◦ Send exe files of builds

20 min	Discussing tools and assets	All	• Mixamo • Speedtree • Substance • Turbosquid

2016-03-02 Sprint Meeting Notes

Date

02 March 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish this sprint
- Setup next sprint

Discussion Items

Time	Item	Who	Notes
10 min	Finish old sprint	All	• Filled in the sprint form
1 h	Created new sprint	All	• What to do this week • Create landscape in test Campaign • Create UI in test campaing • Find menu textures • Research UI • Create a moving unit.

2016-03-09 Sprint Meeting Notes

Date

09 March 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish last sprint
- Set up goals for next sprint

Discussion Items

Time	Item	Who	Notes

45min	Finish last sprint	All	• What we did
			  • Sound settings
			  • Moving characters
			  • Menu textures
			  • AI environment variables
			• Bendik problems with unreal, probably needs to rebuild

40min	Discussion with Mariusz	Mariusz Nowostawski	• Discuss meeting with sarapta • Controlling characters • AI controller
30min	Start new sprint	All	• Further implement characters • Switching turns • Implementing AI • Work further on terrain streaming and border between regions.

2016-03-16 Sprint Meeting Notes

Date

16 March 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish the old sprint
- Start New sprint

Discussion Items

Time	Item	Who	Notes
35min	Finish old sprint	All	• Fill in Sprint log • Close sprint
1h 10min	Setup New Sprint	All	• More advanced AI • Work on Terrain • Improve Character

2016-03-30 Sprint Meeting Notes

Date

30 March 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish old sprint
- Create new sprint

Discussion Items

Time	Item	Who	Notes
1h	Discussing progress	Bendik Hillestad Bjørn Nødland Fuglestad	• Stats system • Weapons • Map
50 min	Finishing sprint	All	• Write sprint log • Complete sprint
20 min	mariusz meeting	All	• Updating marius on progress
15 min	Creating new sprint	All	• Create sprint in JIRA • Fill in goals in sprint log

2016-04-06 Sprint Meeting Notes

Date

06 April 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish old sprint
- Start new sprint

Discussion Items

Time	Item	Who	Notes
20 min	Discuss progress from old sprint	All	• AI progress • Terrain and asset progress • UI and information improvements
20 min	Complete old sprint	All	
35 min	Start new sprint	All	

2016-04-13 Sprint Meeting Notes

Date

13 April 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish old sprint
- Start new sprint

Discussion Items

Time	Item	Who	Notes
1h	Finish old sprint	All	• What we did

1h	Meeting with mariusz	All	• Questions
			• Showing progress
			• Talking about bachelor
			  ◦ Report
			  ◦ Presentation
			• Demo
			• Testing

30 min	Starting new sprint	All	

2016-04-20 Sprint Meeting Notes

Date

20 April 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish old sprint
- Start new sprint

Discussion Items

Time	Item	Who	Notes
45m	Finish old sprint	All	• Integration is delayed. • Nearly done setting up the material blueprints for the terrain textures. • Finished victory condition. • Created a demo for the skirmish combat • Implemented damage types • Added animations to humanoid units and made it so that animations trigger • Created a static load screen and a win and lose screen. • Did not finish ranged attacks and sounds on animations.
45m	Start new sprint	All	• Continue. (Bendik & Per-Arne) • Push campaign map (Bendik) • Add AI damage types. (Per-Arne) • Make units turn towards the unit they are attacking (Bjørn) • Implement a combat log (Bjørn) • Make units not able to attack through walls (Bjørn) • Continue to try to add Ranged units (Bjørn) • Look for sound assets to use (Bjørn) • Start writing on bachelor (All) • Create a new branch off of the dev-branch and then try and merge the campaign map content into it. • Combat log will most likely be similar to how i implemented damage numbers. • Most likely use a ray trace to do ranged attacks, this will most likely also fix the attacking through walls as well.

2016-04-27 Sprint Meeting Notes

Date

27 April 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish old sprint
- Start new sprint

Discussion Items

Time	Item	Who	Notes

35min	Old sprint	All	• What we did last sprint
			  • Push Campaign map
			  • Better attacks
			  • Combat log
			  • Ranged attacks
			  • Better attack hit checks

20min	Supervisor meeting	Mariusz Nowostawski	• Showing progress • Asking about bachelor • Setting up review dates

40min	New Sprint	All	• What to do this sprint
			  • Improvements
			  • Balancing
			  • Continue to merge
			  • More focus on bachelor

2016-05-04 Sprint Meeting Notes

Date

04 May 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish old sprint
- Start new sprint

Discussion Items

Time	Item	Who	Notes

20min	Finish old sprint	All	• What has been done this week
			  • Almost done with AI
			  • Added a lot of improvements to the game based on feedback
			  • Pushed campaign map
			  • Fixed lighting issue
			  • Reduced repo size
			  • Working on bachelor report

20min	Creating new sprint	All	• This week
			  • Continue on unfinished issues from last week
			  • Add a tutorial
			  • More improvements

2016-05-11 Sprint Meeting Notes

Date

11 May 2016

Attendees

- Bjørn Nødland Fuglestad
- Bendik Hillestad
- Per-Arne Waaler Stenshagen

Goals

- Finish old sprint
- Start new sprint

Discussion Items

Time	Item	Who	Notes

20min	Finish old sprint	All	• Was done
			  • Finished graphical bugs
			  • Started working on improvements
			  • Almost ready with AI

20min	Start new sprint	All	• To Do
			  • Finish all relevant issues
			  • Do some play testing
			  • Make gameplay videos

H Sprint Logs

The following pages contain all sprint logs in PDF format. These are divided into individual sprints with subsections. The subsections are:

- Goals
  - Sprint goal of each member.
  - Usually specified to which member they apply. If unspecified, they apply to all.
- Plan
  - Implementation plan for each goal.
  - Structured in same way as Goals section, meaning for example that the second plan written correlates with second goal written. Certain Plans may not correlate perfectly with Goals, in which case it is necessary to take a closer look and see what Plans fit with which Goals.
- Result
  - Results for each goal.
  - Same build as with Plan.
- Retrospective
  - Quick side-note for each sprint.
  - Usually too trivial for thesis, yet too important to ignore.

Sprint logs

Initializer sprint 03 - 10 feb

Goals

- Complete design document to such an extent that the initialization of the game and programming may be started next sprint.
- To create an index page for our hig website. (Bjørn)
- Create sections and subsections of latex document. (Bjørn)

Plan

- Discuss game design ideas and compare notes.
- Use html and css to create a proper home page for the project.
- Look at other bachelor theses to see what sections and subsections to use in our thesis.

Result

- Decided on enough design to start working on implementing the game for the next sprint.
- Created a template/outline for all the pages within our projects web area. Need to fill in some content.
- The sections where quite consistent among the other bachelor theses so it was easy to add them. The problem is that many of the subsections are different from project to project so we will have to see what will fit our project in the future. Also split the thesis into several sub files to make them easier to maintain.

Retrospective

- Good to have asked Mariusz for help, as it has given us new ideas for our game.

Setup Sprint 10 - 17 feb

Goal

- Research terrain creation in unreal (Bendik)
- Create main menu (Bjørn)
- Create settings menu (Bjørn)
- Find assets (Per-Arne)
- Draw up some maps (Per-Arne)

Plan

- Do some googling and reading the documentation to determine the appropriate way to load, generate and display the maps.
- Look at the documentation for coding in unreal. Then use the tutorial for creating menus in the unreal tutorials. If still having problems google for answers and check stack overflow.
- Research amongst several free sound and free music sites, as well as looking thoroughly through the Unreal store and other possible asset sites.
- Try to base some maps on existing ones, as well as make some more “random” maps using Microsoft Paint, or, possibly, other similar drawing tools.

Result

- Done with preliminary research. Got level streaming to work, need to do more research to determine the best way to implement this with our game.
- Finished the main menus functionality. Will need some design changes later.
- Did not get to implement the settings menu. Will be simpler as i now know how user interfaces in unreal works.
- Problems with my (Per-Arne) laptop, pushed to next sprint.
- Made a few maps which can be used as “inspiration” later on.

Retrospective

- Doing level loading/streaming in C++ is a nightmare, and the developers advise against it, so I went with blueprints instead.
- Took longer time than expected because we needed to set ourselves into the unreal engine.
- Should install Unreal and other necessary programs on my (Per-Arne) desktop, to avoid stalls when there is an issue with my laptop (which happens increasingly often).

A New Hope Sprint 17 - 24 feb

Goal

- General research and learning on how to use Unreal Engine (All).
- Continuation on working with Setup Menu widgets (Bjørn).
- Instance/ skirmish/ custom battle menu (Bjørn).
- Conquest/ grand campaign menu (Bjørn).
- Campaign/ story menu (Bjørn).
- Continuation on finding and importing assets (Per-Arne).
- Further research on usage of terrain, level loading and maps in Unreal Engine (Bendik).

Plan

- Individual learning.
- Already know how to create the menus. Will have the most challenge with creating logic for buttons and such. Will most likely be able to find tutorial on how to do it.
- Research amongst several free sound and free music sites, as well as looking thoroughly through the Unreal store and other possible asset sites.
- Go through documentation to find the appropriate tools to use for level streaming and map composition, and search for recommendations on optimal usage of these. Finally, create a test level to confirm that the streaming works as expected.

Result

- Discovered more methods for simplifying level streaming for certain scenarios, which seem applicable for the overworld streaming. Will still need to make a test map to confirm and do some benchmarking.
- Found assets, but will wait with importing them until needed.
- Finished layout for settings menu and campaign menu, did not finish instance and conquest menu.

Retrospective

- A bit early to do instance and conquest menu.
- A bit early to import assets.

The Empire Strikes Back Sprint 24 feb - 02 mar

Goal

- Make a tile prototype to test how it works. (Bjørn)
- Transfer campaign options to campaign level. (Bjørn)
- Move things from the project plan to the LaTeX document. (Bjørn)
- Make a temporary overworld map to test the different methods for level streaming. (Bendik)
- Research AI in Unreal Engine. (Per-Arne)
- Prepare for Sarepta meeting. (Per-Arne)

Plan

- Look at example projects.
- Try and implement the features in the engine.
- Read, watch walkthroughs and try tutorials.
- Make sure everything is in order in front of the meeting, and write down what we need.

Result

- A tile system already exists for purchase.
- Was able to transfer difficulty from menu into a testlevel.
- Moved what was relevant from the project plan into latex thesis.
- Got level streaming to work as well as zooming into the map to reveal detailed structures.
- Most Unreal AI can be done in blueprints, etc.
- Meeting went well, will contact Sarepta on Skype in the future.

Retrospective

- Level streaming in Unreal is a bit weird to use initially since you need 4 different windows open and connect stuff together, but works quite well when it's finished.

Return of the Jedi Sprint 02 mar - 09 mar

Goal

- Integrate the level streaming into to the campaign project (Bendik)
- Create landscape for test campaign (Bendik)
- Create a sub level for a region (Bendik)
- Create a unit that can be moved
- Create a grouped unit
- Create units that can melee attack
- Create units that can range attack
- Create region borders (Bendik?)
- Implement sound options (Bjørn)
- Add content to project webpage (Bjørn)
- Create Campaign UI (Bjørn)
- Create Menu textures (Per-Arne)
- Advanced AI research (Per-Arne)

Plan

- Setup a level in the campaign that implements the level streaming, and make a base terrain for it.
- Already have an idea of how to set the sound settings
- Just need to write how far we have come
- Do the same that i did with the main menu. Will need to do some restructuring of the base classes for the player controller.
- General research on textures, how to make them, and how to implement them.
- General research on more advanced AI, how to sort it, and how to make it seem "alive".

Result

- Made a terrain for the campaign and a subregion for it with some dynamically loaded geometry.
- Could not integrate with the campaign in the dev-branch, will possibly need to rebuild the solution to resolve some references.
- Was able to make a moving unit through blueprints.
- Creating a grouped unit turned out to be harder than expected, was not able to do it in this sprint.
- Was able to make a unit attack another unit within range.
- Did not have time to make a ranged unit, though it could be done by increasing the range of the melee unit.
- Was able to implement sound options.
- Added more content to the project web page, might take some time before it recognizes the changes.
- Made a simple UI to be used in the campaign level.
- Base menu textures setup, but not implemented or uploaded. Will do this in the coming week.
- Figured out how to get AI to look at the environment and use this knowledge to its advantage.

Retrospective

- Might have to reconsider group units.
- Still need to implement more befor the UI is really required.

The Phantom Menace Sprint 09 mar - 16 mar

Goal

- Setup up basic AI for a single character (Per-Arne)
- Continue on the terrain, adding more detail and regions. (Bendik)
- Strengthen the interaction between the AI controller and Base unit (Bjørn)
- Fully implement the base unit character. (Bjørn)
- Try to make it possible to end each player's turn (Bjørn)

Plan

- Start with basic movement and attack, and then expand from there.
- Extend on the already built terrain using level streaming, and finalise the integration with the campaign.
- Move all commands to the character into the unit controller. Research how to talk to character from controller.
- Find out what values that are common for all units and add them to the base unit.

Result

- AI can now move around on its own, track the player, chase the player and track (but not chase) multiple targets.
- Integration of level streaming completed.
- Did not get around to strengthening the integration between AI and Base unit.
- Finished making the base character. Still feel that it can be made more advanced and more polished.
- The player has the ability to end his turn but the AI cannot end its turn.

Retrospective

- There is a lot of ways that the Base unit can be implemented feel that there needs to be more time discussing the different options.

Attack of the Clones Sprint 16 mar - 30 mar

Goal

- Improving the AI to track Allies, choose which of its enemies it should target, attack, and setup framework for using special items/power ups/attacks. (Per-Arne)
- Try and implement borders and continue on terrain. (Bendik)
- (Create a minimap)
- Redesign the Unit UI so that it is more appealing. (Bjørn)
- Change how the attacks work. (Bjørn)
- Movement costs AP. (Bjørn)
- Implement key bindings. (Bjørn)
- Run a performance test. (Bjørn)

Plan

- Extend existing AI to implement new features.
- Add borders as a separate level that hovers above the terrain.
- Work more on the terrain (Use Per-Arne's drawing?), and add more assets.
- Make the units work more like Divinity original sin.

Result

- Borders can be made with splines and a little blueprint. Works fairly well, just a lot of right-clicking.
- Made a terrain based on the drawing by Per-Arne. May have to scale it down slightly, but that's trivial. Still have to add snow and mountain textures, as well as foliage.
- Looked at different ways to create a minimap, and found two easy methods that may work if we're just going for a simple minimap that shows a top-down view without much tactical information.

- Changed how unit stats works. Now it is much more similar to what you would expect from an RPG, with strength, dex and int.
- The attacks are now weapons.
- Changed the unit UI to a more appealing style.
- Did not get around to AP cost, keybinding and performance test because of how long remaking the stats took.
- AI can now track Allies and enemies, but using this to their advantage is still unimplemented, and will be done next week.

Retrospective

- Might have gone overboard on the stats.
- Maybe 4k * 4k resolution on the terrain is too much.

Revenge of the Sith Sprint 30 March - 6 April

Goal

- Make AI cooperate and attack common enemy. (Per-Arne)
- Add foliage and terrain textures, maybe a village? (Bendik)
- Make actions cost action points on units. (Bjørn)
- Find a way to calculate hit chance. (Bjørn)
- Display information such as distance from unit selected under mouse. Range of attacks. AP cost to move distance. Hit chance. Basicly display combat information (Bjørn)
- Create a unit spawner (Bjørn)

Plan

- Continue working from last week.
- Display information similar to divinity original sin.

Result

- AI cooperation done, and framework for strength and weakness exploitation by AI done.
- Tested different types of foliage and terrain textures. Most of the ones from the Kite demo have really high RAM requirements in the editor, but I've found some grass and tree assets that work. I've put together parts of a village / outpost, but it's not complete. Still need to find some mountain and snow assets that work.
- Implemented AP cost for attack and moving, implemented hit chance. Added various methods of displaying information, such as distance, AP cost, damage, health and navigation path.

Retrospective

-

The Force Awakens 06 April - 13 April

Goal

- Complete unfinished items. (All)
- Create melee units and weapons. (Bjørn)
- Add victory conditions. (Bjørn)
- Fix select on player. (Bjørn)
- Improve skirmish controller. (Bjørn)
- Attempt to fix AI bugs. (Per-Arne) (Details in result)

Plan

- Continue work from last week (All).

Result

- Created several weapon blueprints. Created several human and monster units, the monster units have animations such as move, attack, take damage and death. There is currently no animations for human characters as we are missing proper animation assets.
- Fixed select on player.
- Skirmish controller now is mainly c++ code and fixed several small bugs.
- Victory condition for skirmish battles is almost done. There is a problem with it not counting the existing unit that are alive correctly so when that is done it should be possible to play a game.
- Added minimap.
- Looked at some terrain assets at the Unity asset store, bookmarked some of them but haven't imported them into Unreal yet because they all require a lot of manual boring setup.
- Did some polish to one of the "village" areas, trying to focus on "minimum viable product".
- AI-bugs fixed:
    - Bug where all characters ganged up on one enemy resolved. AI should now be able to target multiple enemies.
    - AI refusing to stop moving despite being "close enough" to enemy fixed. AI now stops a certain (adjustable) distance from enemy.
    - Temporary bug where AI tried to attack without it being implemented fixed.

Retrospective

- Bug fix that allows for AI to stop an adjustable distance away from the enemy may also be used for ranged weapons, so a temporary (and annoyingly similar to meelee weapon system) AI system for ranged weapons could be deleted altogether.

- Minimap is quite straightforward, just need a camera and SceneCapture2D component which renders to a render target material, and then display it with a UI widget. Should be able to use a similar setup for a portal if I go that path.

Knights of the Old Republic 13 April - 20 April

Goal

- Successfully merge AI-branch and dev-branch (there are *a lot* of merge conflicts). (Per-Arne)
- Finish victory condition (Bjørn)
- Create a simple demo for the skirmish combat (Bjørn)
- Implement damage types (Bjørn)
- Animations for humanoid units (Bjørn)
- Add sounds to animations (Bjørn)
- Implement ranged attacks (Bjørn)
- Load screen and win and lose screen (Bjørn)
- Continue. (Bendik)

Plan

- Copy everything that conflicts with the merge locally, then paste it straight back into the dev-branch after the merge.
- Most likely need to create a new struct from DamageEvent
- Use animation blueprints for both animations and sound
- Try to do some ray tracing to check for anything blocking, then make an arrow actor.
- Same plan as last time

Result

- Integration is delayed.
- Nearly done setting up the material blueprints for the terrain textures.
- Finished victory condition.
- Created a demo for the skirmish combat
- Implemented damage types
- Added animations to humanoid units and made it so that animations trigger
- Created a static load screen and a win and lose screen.
- Did not finish ranged attacks and sounds on animations.

Retrospective

- Should have merged branches more often and sooner.
- Materials are complicated.
- Loading screens and damage types were more complicated than originally thought.

Republic Commando 20 April - 27 April

Goal

- Continue. (Bendik & Per-Arne)
- Push campaign map (Bendik)
- Add AI damage types. (Per-Arne)
- Make units turn towards the unit they are attacking (Bjørn)
- Implement a combat log (Bjørn)
- Make units not able to attack through walls (Bjørn)
- Continue to try to add Ranged units (Bjørn)
- Look for sound assets to use (Bjørn)
- Start writing on bachelor (All)

Plan

- Create a new branch off of the dev-branch and then try and merge the campaign map content into it.
- Combat log will most likely be similar to how i implemented damage numbers.
- Most likely use a ray trace to do ranged attacks, this will most likely also fix the attacking through walls as well.

Result

- Pushed campaign map
- Was sick at beginning of sprint, and made several bad judgments and errors, including linking faults and reference errors. This was fixed when I got better, but slowed me down considerably. Need to continue work next sprint.
- Units now face the direction that they are attacking
- There is now a combat log
- Units now do a line trace to check for line of sight when attacking
- Added ranged units to the game. Projectiles might hit friends and environment
- Looked for some sound assets, might add them.

Retrospective

- Working on blueprints while sick is not a good idea.

Empire At War 27 April - 4 May

Goal

- Upload AI and merge with dev-branch. (Per-Arne)
- Implement improvements based on feedback (Bjørn)
- Balance units and weapons (Bjørn)
- Reduce repository size (Bendik)

- Resolve lighting issues (Bendik)
- Add battle selection campaign (Bendik)
- Work on bachelor report (All)

Plan

- Continue as before.
- From monday work working 10 - 16 on bachelor report.

Result

- Trying to implement AI to follow multiple unique actors at once, but is proving to be difficult. The rest of the AI is done, but not yet uploaded.
- Most of the improvements based on feedback have been fixed.
- Started working on balancing, will take more time.
- Repository size and lighting issues resolved.
- Battle selection not yet implemented.
- Bachelor report is now about 30 pages without appendix.

Retrospective

- Bitbucket can be very slow. Used a day to perform garbage collection on deleted files in the repo.

Rogue One 4 May - 11 May

Goal

- Continue working on bachelor report (All)
- Implement battle selection (Bendik)
- Continue to balance the game (Bjørn)
- Second iteration of improvements (Bjørn)
- Clean code (Bjørn)
- Add video tutorial to the game (Bjørn)
- Continue to improve and merge AI (Per-Arne)

Plan

- Go through the code and improve structure, try and create a lint to check for code style errors. Also make sure that the code is well documented.
- Record some gameplay while explaining things using shadowplay. Then play video when tutorial button is pressed.
- Continue as before

Result

- Battle selection isn't quite ready yet.
- Did a round of balancing. Most likely not perfect but better than before.

- Did implement several improvements to the code, levels and blueprints.
- Did not get around to adding a tutorial, will have to continue other unfinished tasks in the next sprint.
- Workarounds and the majority of implementation done, will be completed and merged by Sunday 15.05.2016.

Retrospective

- Writing the bachelor report is taking up a lot of time and slowing down progress on the game.

The Saga Ends 11 May - 17 May

Goal

- Finish Bachelor report (All)
- Do some testing with classmates (All)
- Finish battle selection (Bendik)
- Finish AI merge (Per-Arne)
- Finish tutorial (Bjørn)
- Finish cleaning code (Bjørn)
- Finish the last batch of improvements (Bjørn)
- Create some gameplay videos (Bjørn)

Plan

- Finish up most of the report Wednesday-Friday and Monday, complete it and hand it in on Tuesday 17. May.
- Have students play the game in the game lab, prepare some questions.
- Use the weekend to finish up the features.

Result

- Bachelor report being finished now.
- Battle selection still not quite ready, will be completed after delivery of the report.
- AI finally merged, but with certain files missing. Gained several Software issues while trying to re-stage these files (Source Tree won't pull or clone, Unreal Engine 4 won't initialize, One Drive crashed), but as the restaging itself is a small job, it should be up and running quickly when solutions to the Software issues are found.
- Added tutorial
- Finished cleaning code
- Finished last small improvements
- Added some gameplay videos to youtube

Retrospective

- A lot of Software issues can happen in a surprisingly short amount of time.

I Jira Issues

The following pages contain issues and sprints created and updated using Atlassian Jira converted to PDF format. The majority of these issues correlate to the Sprint Logs H.

[NOMEM-100] Fix final AI issues Created: 17/May/16 Updated: 17/May/16	
Status:	In Progress
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Bug	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Unresolved	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Sprint:	Final touches
Story Points:	12

[NOMEM-99] Create gameplay videos Created: 11/May/16 Updated: 17/May/16	
Status:	In Progress
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Unresolved	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Sprint:	The Saga Ends, Final touches
Story Points:	4

Description

Create some gameplay video of the game that the readers of the bachelor can check out.

[NOMEM-97] Testing with classmates Created: 11/May/16 Updated: 17/May/16 Resolved: 17/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 hour		
Original Estimate:	Not Specified		

Sprint:	The Saga Ends
Story Points:	3

Comments

Comment by Per-Arne Waaler Stenshagen [17/May/16]
Had people we know test it.

[NOMEM-96] Work on Bachelor report

Created: 11/May/16 Updated: 17/May/16 Due: 17/May/16 Resolved: 17/May/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 weeks, 3 days, 4 hours		
Original Estimate:	Not Specified		

Sprint:	The Saga Ends
Story Points:	42

Comments

Comment by Per-Arne Waaler Stenshagen [17/May/16]

Almost done

[NOMEM-95] Work on Bachelor report Created: 11/May/16 Updated: 17/May/16 Due: 17/May/16 Resolved: 17/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 weeks, 3 days, 4 hours		
Original Estimate:	Not Specified		

Sprint:	The Saga Ends
Story Points:	42

Comments

Comment by Per-Arne Waaler Stenshagen [17/May/16]
Just hand-in , minor fixes and appendices remaining.

[NOMEM-94] AI fix ability to track multiple characters

Created: 04/May/16 Updated: 15/May/16 Resolved: 15/May/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Bug	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 3 hours		
Original Estimate:	Not Specified		

Sprint:	Rogue One, The Saga Ends
Story Points:	6

Comments

Comment by Per-Arne Waaler Stenshagen [15/May/16]

Seems to be working, may have issues

[NOMEM-93] Clean code Created: 04/May/16 Updated: 15/May/16 Resolved: 15/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 4 hours		
Original Estimate:	Not Specified		
Environment:	C++		

Sprint:	Rogue One, The Saga Ends
Story Points:	4

Description

Ensure that the code is clean well documented and readable.

Comments

Comment by Bjørn Nødland Fuglestad [15/May/16]

Cleaned code to meet code convention and documentation syntax

[NOMEM-92] More Improvements Created: 04/May/16 Updated: 17/May/16	
Status:	In Progress
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Unresolved	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		
Environment:	C++, Blueprint		

Sprint:	Rogue One, The Saga Ends, Final touches
Story Points:	3

Description

Finish some of the improvements that have been waiting to be added.
1. Fix skirmish ui class
2. Attack damage range show in UI
3. Nav mesh visible when within range
4. Remove buttons on action bar or collapse them

[NOMEM-91] Create a tutorial video Created: 03/May/16 Updated: 17/May/16 Resolved: 17/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	3 hours		
Original Estimate:	Not Specified		

Sprint:	Rogue One, The Saga Ends, Final touches
Story Points:	3

Description

Create a video to use instead of a tutorial level. When the user presses the tutorial button simply play the video.

Comments

Comment by Bjørn Nødland Fuglestad [17/May/16]

Made a UI widget that plays a video material and plays the associated audio.

[NOMEM-90] Select battles in campaign Created: 27/Apr/16 Updated: 17/May/16	
Status:	In Progress
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bendik Hillestad
Resolution:	Unresolved	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Sprint:	Empire At War, Rogue One, The Saga Ends, Final touches
Story Points:	5

Description

Make it so that the player can select battles in campaign game mode

[NOMEM-89] Resolve lighting issues. Created: 27/Apr/16 Updated: 29/Apr/16 Resolved: 27/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Bug	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 hour		
Original Estimate:	Not Specified		

Sprint:	Empire At War
Story Points:	3

[NOMEM-88] Reduce repository size. Created: 27/Apr/16 Updated: 29/Apr/16 Resolved: 29/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 hours		
Original Estimate:	Not Specified		

Sprint:	Empire At War
Story Points:	2

Description

Reduce repo size.

Comments

Comment by Bendik Hillestad [29/Apr/16]

Took some wrestling with Bitbucket, but size has now been reduced

[NOMEM-87] Balance units and weapons Created: 27/Apr/16 Updated: 07/May/16 Resolved: 07/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	7 hours		
Original Estimate:	Not Specified		

Sprint:	Empire At War, Rogue One
Story Points:	3

Description

Balance units and weapons

Comments

Comment by Bjørn Nødland Fuglestad [07/May/16]

Balancing the units and weapons. Will most likely need to be done again.

[NOMEM-86] Clean Design Document Created: 26/Apr/16 Updated: 17/May/16 Resolved: 17/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 hour		
Original Estimate:	Not Specified		
Environment:	Google doc		

Sprint:	Empire At War, Rogue One, The Saga Ends
Story Points:	3

Description

The design doc is in a really bad state. Fill in information and clean up.

Comments

Comment by Per-Arne Waaler Stenshagen [17/May/16]

Ready for upload

[NOMEM-85] Improve game based on feedback from Amund Created: 24/Apr/16 Updated: 02/May/16 Resolved: 02/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 days, 1 hour		
Original Estimate:	Not Specified		
Environment:	c++, Blueprint		

Sprint:	Empire At War
Story Points:	5

Description

Fix the issues that Amund pointed out in his e-mail.
1. Fix camera.
2. Make arrow keys also fire same input as wasd.
3. Make the keybindings visible even if not implemented.
4. Middle mouse move camera (optional).
5. Check end game condition on unit death.
6. More information when its your turn (Big display in widget)
7. Indicate no more action points.
8. Move to attack (Possibly its own issue).
9. Red path when none is valid
10. Better indication of friend and foe (Possibly its own issue).
11. Highlight enemies within range.
12. Fix spelling errors and inconsistency (see email).
13. Disable or fix difficulty
14. Select ring under monsters are hard to see.
15. Move difficulty to the other side of the image in skirmish and campaign menu
16. Better messaging when starting without choosing a level.

Comments

Comment by Bjørn Nødland Fuglestad [02/May/16]

Finished most of the improvements that where specified. Still need some changes but they will most likely be done in another issue.

[NOMEM-83] Work on the bachelor report

Created: 20/Apr/16 Updated: 17/May/16 Due: 17/May/16 Resolved: 17/May/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Critical
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	1 week, 1 day, 4 hours		
Time Spent:	2 weeks, 3 days, 4 hours		
Original Estimate:	4 weeks		

Sprint:	Republic Commando, Empire At War, Rogue One, The Saga Ends
Story Points:	42

Description

Work on filling in the Bachelor report.

Comments

Comment by Per-Arne Waaler Stenshagen [17/May/16]

Almost done

[NOMEM-82] Make units not able to attack through walls Created: 20/Apr/16 Updated: 21/Apr/16 Resolved: 21/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		

Sprint:	Republic Commando
Story Points:	2

Comments

Comment by Bjørn Nødland Fuglestad [21/Apr/16]

Added a line trace when checking if possible to attack, now other units and walls block attacks.

[NOMEM-80] Make units turn towards the unit they are attacking Created: 20/Apr/16 Updated: 20/Apr/16 Resolved: 20/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Minor
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 hour		
Original Estimate:	Not Specified		

Sprint:	Republic Commando
Story Points:	1

Comments

Comment by Bjørn Nødland Fuglestad [20/Apr/16]
Finished making attacking unit face defending unit.

[NOMEM-79] Add AI damage types Created: 20/Apr/16 Updated: 04/May/16 Resolved: 04/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	Republic Commando, Empire At War
Story Points:	4

Comments

Comment by Per-Arne Waaler Stenshagen [04/May/16]
Could be expanded, but good enough for now.

[NOMEM-78] Push campaign map Created: 20/Apr/16 Updated: 29/Apr/16 Resolved: 27/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day		
Original Estimate:	Not Specified		

Sprint:	Republic Commando
Story Points:	4

Comments

Comment by Bjørn Nødland Fuglestad [27/Apr/16]
Works in editor, but refuses to compile
Comment by Bendik Hillestad [29/Apr/16]
Now compiles properly

[NOMEM-77] Add damage types Created: 13/Apr/16 Updated: 19/Apr/16 Resolved: 19/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 4 hours		
Original Estimate:	Not Specified		
Environment:	C++		

Sprint:	Knights of the Old Repuplic
Story Points:	4

Description

Add different damage types such as fire, earth, air, blunt, slash and so on.

Comments

Comment by Bjørn Nødland Fuglestad [19/Apr/16]

Created some basic damage classes that each effect the damage done to a a character indifferent ways.

[NOMEM-76] Create a Demo Created: 13/Apr/16 Updated: 14/Apr/16 Resolved: 14/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	Knights of the Old Repuplic
Story Points:	2

Description

Create a demo to that can be played on mac and on windows showing off the skirmish map battles.

Comments

Comment by Bjørn Nødland Fuglestad [14/Apr/16]

Finished editing code for the project so that it would package.

[NOMEM-75] Load Screen Created: 13/Apr/16 Updated: 16/Apr/16 Resolved: 16/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	3 hours		
Original Estimate:	Not Specified		
Environment:	Blueprint, C++		

Sprint:	Knights of the Old Repuplic
Story Points:	2

Description

Create a loading screen when starting skirmishes or campaigns.

Comments

Comment by Bjørn Nødland Fuglestad [16/Apr/16]

Made a static load screen when changing to campaign and skirmish levels from main menu. To have a animated loading screen level streaming or placing animation on its own thread is needed.

[NOMEM-74] Merge AI-branch with dev-branch Created: 13/Apr/16 Updated: 15/May/16 Due: 08/May/16 Resolved: 15/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	3 weeks, 1 hour		
Original Estimate:	Not Specified		

Sprint:	Knights of the Old Repuplic, Republic Commando, Empire At War, Rogue One, The Saga Ends
Story Points:	7

Comments

Comment by Per-Arne Waaler Stenshagen [15/May/16]
Should be up and running now

[NOMEM-73] Add sounds to units Created: 08/Apr/16 Updated: 15/May/16 Due: 30/Apr/16 Resolved: 14/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	3 hours		
Original Estimate:	10 minutes		
Environment:	C++ and blueprints		

Sprint:	Knights of the Old Repuplic, Republic Commando, The Saga Ends
Story Points:	3

Description

Find some sound assets to be used for when taking damage, attacking, walking and maybe some for the troll idle.

Comments

Comment by Bjørn Nødland Fuglestad [14/May/16]

Added sound to inits attack and hit damage.

[NOMEM-72] Add animations to humanoid units

Created: 08/Apr/16 Updated: 17/Apr/16 Due: 30/Apr/16 Resolved: 17/Apr/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	3 hours		
Time Spent:	1 day, 1 hour		
Original Estimate:	1 day, 4 hours		
Environment:	C++ and blueprints		

Sprint:	Knights of the Old Repuplic
Story Points:	6

Description

Find animations for the non human units and set up the animations. Also add the needed code for the different states to trigger.

Comments

Comment by Bjørn Nødland Fuglestad [17/Apr/16]

All units now have idle/walk, attack, hit and death animations. All get triggered accurately during game play, though it could always look better.

[NOMEM-71] Add victory condition Created: 06/Apr/16 Updated: 14/Apr/16 Resolved: 14/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 days, 2 hours		
Original Estimate:	Not Specified		

Sprint:	The Force Awakens, Knights of the Old Repuplic
Story Points:	5

Comments

Comment by Bjørn Nødland Fuglestad [14/Apr/16]
Created a victory condition for skirmish game mode that is checked at the end of each turn.

[NOMEM-70] Fix select on player Created: 06/Apr/16 Updated: 09/Apr/16 Resolved: 09/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Bug	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	5 hours		
Original Estimate:	Not Specified		

Sprint:	The Force Awakens
Story Points:	3

Comments

Comment by Bjørn Nødland Fuglestad [09/Apr/16]
Fixed decal visibility, re-positioned decals in base character and in monsters. Added a range decal. Color coded decals based on team. Added circle decals of different sizes.

[NOMEM-69] Improve skirmish controller Created: 06/Apr/16 Updated: 11/Apr/16 Resolved: 11/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 days		
Original Estimate:	Not Specified		

Sprint:	The Force Awakens
Story Points:	6

Comments

Comment by Bjørn Nødland Fuglestad [11/Apr/16]
Finished converting Skirmishplayer to c++, also made general improvements along the way.

[NOMEM-68] Create 10 melee units and weapons Created: 06/Apr/16 Updated: 08/Apr/16 Resolved: 08/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 days		
Original Estimate:	Not Specified		

Sprint:	The Force Awakens
Story Points:	6

Comments

Comment by Bjørn Nødland Fuglestad [08/Apr/16]

Created several human units and monster units along with weapons for each. Placed each of them into the stat table. Added walk animations for the non human characters as that was possible using the included assets. Will need some more work to get the attack, hit and death animations playing. Will also need some animation assets for the human characters.

[NOMEM-67] Attempt to fix AI bugs Created: 06/Apr/16 Updated: 13/Apr/16 Resolved: 13/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Bug	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	7 hours		
Original Estimate:	Not Specified		

Sprint:	The Force Awakens
Story Points:	4

[NOMEM-66] Create unit spawners Created: 30/Mar/16 Updated: 06/Apr/16 Resolved: 06/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		
Environment:	C++, blueprint		

Sprint:	Revenge of the Sith, The Force Awakens
Story Points:	4

Description

Create a spawner able to spawn different units from a save game object.

Comments

Comment by Bjørn Nødland Fuglestad [06/Apr/16]

Made a simple spawner that, spawns units within its own volume. Still needs some way of getting the units it needs to spawn. Should also spawn then in a pattern rather than random.

Comment by Bjørn Nødland Fuglestad [06/Apr/16]

Finished making a spawner that spawns units from an array of unit classes. Also takes the team to assign them to. They are now spawned in a grid pattern.

[NOMEM-65] Display combat information

Created: 30/Mar/16 Updated: 05/Apr/16 Resolved: 05/Apr/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	3 days		
Original Estimate:	Not Specified		
Environment:	C++		

Sprint:	Revenge of the Sith
Story Points:	8

Description

Display distance between unit and mouse, Display ap cost, display hit chance and display range.

Comments

Comment by Bjørn Nødland Fuglestad [05/Apr/16]

Now displaying move distance and cost, Hit chance, navigation spline, health bar on hover over and damage numbers on hit.

[NOMEM-63] Calculate hit chance Created: 30/Mar/16 Updated: 01/Apr/16 Resolved: 01/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	7 hours		
Original Estimate:	Not Specified		
Environment:	C++		

Sprint:	Revenge of the Sith
Story Points:	4

Description

Find a way to calculate hit chance and other stuff like miss, dodge and block

Comments

Comment by Bjørn Nødland Fuglestad [01/Apr/16]

Implemented calculation of hit chance. Now each character has a hit chance and dodge chance that they use. Weapons also affect the hit chance.

Example: The base hit chance is 40% then we check what attribute the weapon has say strength. We then take the character's strength * a ration and add it to the total hit chance. Say you have 10 strength and the ratio is 6% then we add 60% to the hit chance making it 100%.

The final hit chance is then reduced by the enemies dodge chance.

It will need balancing!

[NOMEM-62] Add a village / villages Created: 30/Mar/16 Updated: 27/Apr/16 Resolved: 27/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 days		
Original Estimate:	Not Specified		

Sprint:	Revenge of the Sith, The Force Awakens, Knights of the Old Repuplic, Republic Commando
Story Points:	4

[NOMEM-61] Add snow, mountains and farmland Created: 30/Mar/16 Updated: 27/Apr/16 Resolved: 27/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day		
Original Estimate:	Not Specified		

Sprint:	Revenge of the Sith, The Force Awakens, Knights of the Old Repuplic, Republic Commando
Story Points:	4

[NOMEM-60] Add foliage Created: 30/Mar/16 Updated: 06/Apr/16 Resolved: 06/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	Revenge of the Sith
Story Points:	4

[NOMEM-59] Combat log Created: 25/Mar/16 Updated: 21/Apr/16 Resolved: 21/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		
Environment:	C++, Blueprint		

Sprint:	Republic Commando
Story Points:	2

Description

Create a combat log for in game battles

Comments

Comment by Bjørn Nødland Fuglestad [21/Apr/16]

Finished adding a combat log that adds new lines of text to its self. Possible to toggle visibility and scroll able.

[NOMEM-58] AI item /special attack framework

Created: 16/Mar/16 Updated: 13/Apr/16 Resolved: 13/Apr/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones, Revenge of the Sith, The Force Awakens
Story Points:	4

Description

Setup initial framework for the usage of items, special attacks or other abilities.

[NOMEM-57] AI add attack Created: 16/Mar/16 Updated: 04/May/16 Resolved: 04/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	5 hours		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones, Knights of the Old Repuplic, Republic Commando, Empire At War
Story Points:	4

Description

Make AI attack enemies, not just chase them.

Comments

Comment by Per-Arne Waaler Stenshagen [04/May/16]

AI attack againts a single actor implemented, but checking multiple actors needs fixing. This problem has been turned into its own issue.

[NOMEM-56] AI Specified enemy targeting Created: 16/Mar/16 Updated: 06/Apr/16 Resolved: 06/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 6 hours		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones, Revenge of the Sith
Story Points:	4

Description

Make the AI decide who to target among several enemies

[NOMEM-55] AI track Allies Created: 16/Mar/16 Updated: 24/Mar/16 Resolved: 24/Mar/16	
Status:	Closed
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones
Story Points:	3

Description

Make the AI figure out where its allies are. To be used later during AI teamwork implementations.

[NOMEM-53] Create a minimap Created: 16/Mar/16 Updated: 13/Apr/16 Resolved: 13/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 hours		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones, Revenge of the Sith, The Force Awakens
Story Points:	4

[NOMEM-52] Do a performance analysis Created: 16/Mar/16 Updated: 24/Apr/16 Resolved: 24/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 hours		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones, The Force Awakens, Republic Commando
Story Points:	1

Description

Perform a performance test.

Comments

Comment by Bjørn Nødland Fuglestad [24/Apr/16]

Figured out the command for generating FPS charts. Have a small problem with excel not wanting to make graphs of the data but all the same. Also recorded some data using the profiler.

[NOMEM-51] Redesign the units and attacks Created: 16/Mar/16 Updated: 29/Mar/16 Resolved: 29/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 week, 1 day, 2 hours		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones
Story Points:	14

Description

Redesign how the stats for the units. Change how the attacks work and further implement them.

Comments

Comment by Bjørn Nødland Fuglestad [29/Mar/16]

Created a stats system similar to what you would expect form an RPG, With strength, int, dex and so on. Changed attacks into weapons and made it possible for weapons to have stats such as + 10 dex and such. All units default values are set using a data table. They only need to know the row name that they are to set the default values.

[NOMEM-50] Implement Action point cost Created: 16/Mar/16 Updated: 31/Mar/16 Resolved: 31/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 1 hour		
Original Estimate:	Not Specified		
Environment:	C++ Blueprints		

Sprint:	Attack of the Clones, Revenge of the Sith
Story Points:	3

Description

Make actions cost Action points.

Comments

Comment by Bjørn Nødland Fuglestad [31/Mar/16]

Make actions such as attack and move cost actions points. Remember to call the move functions in character so that the costs are updated.

[NOMEM-49] Improve the UI for units Created: 16/Mar/16 Updated: 29/Mar/16 Resolved: 29/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 days		
Original Estimate:	Not Specified		
Environment:	C++, Blueprints		

Sprint:	Attack of the Clones
Story Points:	4

Description

Change the UI for units to be more appealing and show more information.

Comments

Comment by Bjørn Nødland Fuglestad [29/Mar/16]

Changed health bar and action bar. Made widget for displaying stats and weapon stats. Made general improvements to skirmish ui, such as pausing the game when in menu.

[NOMEM-48] Continue on the terrain Created: 16/Mar/16 Updated: 30/Mar/16 Resolved: 30/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	3 days		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones
Story Points:	12

[NOMEM-46] Increase the commands available to the AI controller

Created: 09/Mar/16 Updated: 11/May/16 Resolved: 11/May/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day		
Original Estimate:	Not Specified		
Environment:	Blueprints and C++		

Sprint:	The Phantom Menace, Attack of the Clones, Rogue One
Story Points:	4

Description

Increase the amount of commands the ai controller can apply to the character it controls.

Comments

Comment by Per-Arne Waaler Stenshagen [11/May/16]

Several commands ignored, as end of project approaches. Focusing instead on implementing and merging what is necessary.

[NOMEM-45] Switch turns between teams. Created: 09/Mar/16 Updated: 16/Mar/16 Resolved: 16/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Incomplete	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 5 hours		
Original Estimate:	Not Specified		
Environment:	Blueprint and c++		

Sprint:	The Phantom Menace
Story Points:	6

Description

Change which team is able to move their units.

Comments

Comment by Bjørn Nødland Fuglestad [16/Mar/16]

Was able to make the player end his turn but not make the AI end his turn.

[NOMEM-44] Further implement the base of the unit character. Created: 09/Mar/16 Updated: 13/Mar/16 Resolved: 13/Mar/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 days, 4 hours		
Original Estimate:	Not Specified		
Environment:	Blueprint and C++		

Sprint:	The Phantom Menace
Story Points:	2

Description

Implement all the values that are needed for the base of units. Also create the functions to be overwritten by extending classes.

Comments

Comment by Bjørn Nødland Fuglestad [13/Mar/16]

Finished a basic setup for a unit. Will need to be polished in the future.

[NOMEM-43] Continue on the terrain Created: 09/Mar/16 Updated: 16/Mar/16 Resolved: 16/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		

Sprint:	The Phantom Menace
Story Points:	3

[NOMEM-42] Implement basic AI on a character Created: 09/Mar/16 Updated: 19/Apr/16 Resolved: 19/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	The Phantom Menace, Attack of the Clones, Revenge of the Sith, The Force Awakens, Knights of the Old Repuplic
Story Points:	5

Description

Move + attack

Comments

Comment by Per-Arne Waaler Stenshagen [19/Apr/16]

Used this Issue to create an early AI-branch, which has finally been closed.

[NOMEM-41] Implement fog of war in combat Created: 06/Mar/16 Updated: 17/May/16	
Status:	In Progress
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Unresolved	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Sprint:	Final touches
Story Points:	12

[NOMEM-40] Make units able to range attack Created: 02/Mar/16 Updated: 23/Apr/16 Resolved: 23/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 2 hours		
Original Estimate:	Not Specified		

Sprint:	Return Of The Jedi, Knights of the Old Repuplic, Republic Commando
Story Points:	4

Description

Make units able to attack from range

Comments

Comment by Bjørn Nødland Fuglestad [23/Apr/16]

Made it possible to attack with ranged attacks. There is now a mage with a fireball attack on his wand. And added a rock that trolls can throw.

[NOMEM-39] Create region borders Created: 02/Mar/16 Updated: 30/Mar/16 Resolved: 30/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		
Environment:	C++, blueprint		

Sprint:	Return Of The Jedi, The Phantom Menace, Attack of the Clones
Story Points:	3

Description

For the regions in the game give them some way to be easy to tell apart

[NOMEM-38] Make Units able to melee attack Created: 02/Mar/16 Updated: 13/Mar/16 Resolved: 13/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 5 hours		
Original Estimate:	Not Specified		
Environment:	C++, blueprint		

Sprint:	Return Of The Jedi, The Phantom Menace
Story Points:	4

Description

Make units able to melee attack

Comments

Comment by Bjørn Nødland Fuglestad [07/Mar/16]

was able to set up a system in which a unit takes damage when in range of the attacker.

Comment by Bjørn Nødland Fuglestad [13/Mar/16]

Finished adding a basic attack system into the project. Will most likely be extended in the future.

[NOMEM-37] Create sub-level for a region Created: 02/Mar/16 Updated: 09/Mar/16 Resolved: 09/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	Return Of The Jedi
Story Points:	1

[NOMEM-36] Create landscape for test campaign Created: 02/Mar/16 Updated: 09/Mar/16 Resolved: 09/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	Return Of The Jedi
Story Points:	1

[NOMEM-34] Integrate level streaming into the campaign project Created: 02/Mar/16 Updated: 16/Mar/16 Resolved: 16/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		

Sprint:	Return Of The Jedi, The Phantom Menace
Story Points:	1

[NOMEM-33] Create a unit that can be moved

Created: 02/Mar/16 Updated: 09/Mar/16 Resolved: 09/Mar/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 4 hours		
Original Estimate:	Not Specified		
Environment:	C++, blueprint		

Sprint:	Return Of The Jedi
Story Points:	4

Description

Create a character that can be moved by selecting it and right clicking

Comments

Comment by Bjørn Nødland Fuglestad [06/Mar/16]

Finished creating a unit that can be moved in blueprints. have still not transferred it to the main project and changed it to c++ code.

Comment by Bjørn Nødland Fuglestad [08/Mar/16]

Transferred the code from side project.

Comment by Bjørn Nødland Fuglestad [09/Mar/16]

Moved the implementation into the dev branch. Still need to make changes to the inheritance and implementation of units, but the idea is still there.

[NOMEM-32] Create Campaign UI Created: 02/Mar/16 Updated: 03/Mar/16 Resolved: 03/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		
Environment:	C++, blueprint		

Sprint:	Return Of The Jedi
Story Points:	1

Description

Make campaign player controller, tweek controller hierarchy

Comments

Comment by Bjørn Nødland Fuglestad [03/Mar/16]

Finished setting up the basic for the campaign UI.

[NOMEM-30] Research advanced AI in Unreal Created: 02/Mar/16 Updated: 09/Mar/16 Resolved: 09/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	3 hours, 30 minutes		
Original Estimate:	Not Specified		

Sprint:	Return Of The Jedi
Story Points:	1

Description

Base Ai tutorials for Unreal where quite basic. More research is needed for an advanced strategy AI.

[NOMEM-29] Fill in content in the project web page

Created: 29/Feb/16 Updated: 03/Mar/16 Resolved: 03/Mar/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 hours		
Original Estimate:	Not Specified		
Environment:	HTML		

Sprint:	Return Of The Jedi
Story Points:	1

Comments

Comment by Bjørn Nødland Fuglestad [03/Mar/16]

Added some content to the web page for the project.

[NOMEM-28] Implement Sound options Created: 29/Feb/16 Updated: 04/Mar/16 Resolved: 04/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	5 hours		
Original Estimate:	Not Specified		
Environment:	Unreal blueprint and C++		

Sprint:	Return Of The Jedi
Story Points:	2

Comments

Comment by Bjørn Nødland Fuglestad [04/Mar/16]

Made it possible to change the sound options. still needs to be saved to file and loaded from file.

[NOMEM-26] Implement Graphic options Created: 29/Feb/16 Updated: 17/May/16	
Status:	In Progress
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Unresolved	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		
Environment:	Unreal 4 blueprint and C++		

Sprint:	Final touches
Story Points:	3

Description

Implement the functionality for the graphics options.

[NOMEM-25] Add Project plan to thesis Created: 24/Feb/16 Updated: 29/Feb/16 Resolved: 29/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day		
Original Estimate:	Not Specified		
Environment:	Latex		

Sprint:	The Empire Strikes Back
Story Points:	2

Description

Move the information from the Google doc into shareLatex thesis.

Comments

Comment by Bjørn Nødland Fuglestad [29/Feb/16]

Copied what i thought was useful from the project plan into the latex thesis. Might need to revise some of it as the past or present tense might be wrong.

[NOMEM-24] Make temporary overworld map for level streaming tests

Created: 24/Feb/16 Updated: 02/Mar/16 Resolved: 02/Mar/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bendik Hillestad
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 7 hours		
Original Estimate:	Not Specified		

Sprint:	The Empire Strikes Back
Story Points:	2

Comments

Comment by Bendik Hillestad [02/Mar/16]

Well, it works!

[NOMEM-23] Reseach AI in Unreal Created: 24/Feb/16 Updated: 02/Mar/16 Resolved: 02/Mar/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		

Sprint:	The Empire Strikes Back
Story Points:	2

Description

Unreal has several AI features, need to figure them out.

[NOMEM-22] Transfer campaign options to campaign level Created: 24/Feb/16 Updated: 26/Feb/16 Resolved: 26/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 1 hour, 30 minutes		
Original Estimate:	Not Specified		

Sprint:	The Empire Strikes Back
Story Points:	2

Comments

Comment by Bjørn Nødland Fuglestad [26/Feb/16]

Finished implementing a way to save setting from campaign menu and load them in campaign level.

[NOMEM-21] Make a tile prototype Created: 24/Feb/16 Updated: 29/Feb/16 Resolved: 29/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Bendik Hillestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Won't Fix	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	The Empire Strikes Back
Story Points:	2

Comments

Comment by Bjørn Nødland Fuglestad [29/Feb/16]

Looked around and found a tool set the does handle tiles. Might consider using it or terrain it has not been decided yet

[NOMEM-20] Implement Keybindings Created: 24/Feb/16 Updated: 17/May/16	
Status:	In Progress
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Unresolved	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Sprint:	Attack of the Clones, Final touches
Story Points:	4

Description

In the keybindings menu make it possible to change keys using the menu.

[NOMEM-19] Campaign Menu Layout Created: 17/Feb/16 Updated: 22/Feb/16 Resolved: 22/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 hours		
Original Estimate:	Not Specified		
Environment:	C++, Unreal, Blueprint		

Sprint:	A New Hope
Story Points:	2

Description

Create the campaign menu.

Comments

Comment by Bjørn Nødland Fuglestad [22/Feb/16]

Created the basic layout. Needs functionality and needs some better textures.

[NOMEM-17] Instance Menu Layout Created: 17/Feb/16 Updated: 20/Apr/16 Resolved: 20/Apr/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 hours		
Original Estimate:	Not Specified		
Environment:	C++, Unreal, Blueprint		

Sprint:	A New Hope, Republic Commando
Story Points:	2

Description

Create the menu for instancing.

Comments

Comment by Bjørn Nødland Fuglestad [24/Feb/16]

Made some of the layout but not all of it.

Comment by Bendik Hillestad [20/Apr/16]

Finished this issue when i was implementing new levels in skirmish mode.

[NOMEM-16] Prepare for Serapta meeting Created: 17/Feb/16 Updated: 02/Mar/16 Resolved: 02/Mar/16	
Status:	Closed
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Blocker
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 hour		
Original Estimate:	Not Specified		

Sprint:	The Empire Strikes Back
Story Points:	1

[NOMEM-15] More reasearch on how to use Unreal Engine

Created: 17/Feb/16 Updated: 24/Feb/16 Resolved: 24/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		

Sprint:	A New Hope
Story Points:	2

Comments

Comment by Bendik Hillestad [24/Feb/16]

Done figuring out stuff, will figure out more.

[NOMEM-14] More reasearch on how to use Unreal Engine

Created: 17/Feb/16 Updated: 24/Feb/16 Resolved: 24/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bendik Hillestad
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	A New Hope
Story Points:	2

Comments

Comment by Bendik Hillestad [24/Feb/16]

Done more reading and fooling around with the engine.

[NOMEM-13] More reasearch on how to use Unreal Engine Created: 17/Feb/16 Updated: 24/Feb/16 Resolved: 24/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	4 hours		
Original Estimate:	Not Specified		

Sprint:	A New Hope
Story Points:	2

Comments

Comment by Bendik Hillestad [24/Feb/16]
Research transferring variables between levels.

[NOMEM-12] More terrain research/ Create a test map

Created: 17/Feb/16 Updated: 24/Feb/16 Resolved: 24/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bendik Hillestad
Resolution:	Incomplete	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	6 hours		
Original Estimate:	Not Specified		

Sprint:	A New Hope
Story Points:	3

Comments

Comment by Bendik Hillestad [24/Feb/16]

Done more research, and there seems like there's a relatively simple way to do the level streaming for the overworld, but need to make a test map to confirm. Level streaming for the "dungeon" part will likely still need manual level streaming with blueprints.

[NOMEM-11] Terrain setup research Created: 10/Feb/16 Updated: 17/Feb/16 Resolved: 17/Feb/16	
Status:	Closed
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bendik Hillestad
Resolution:	Done	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 days		
Original Estimate:	Not Specified		

Sprint:	Setup Sprint
Story Points:	3

Description

Research how to implement maps and map creation in Unreal Engine.

Comments

Comment by Bendik Hillestad [17/Feb/16]

Initial research has been completed, and I have found that using C++ for this is not recommended and actually ends up just looking like a text version of blueprints minus all the benefits that come with using blueprints (Such as not having to calculate the path to the modules yourself, and getting proper callbacks upon stream completion). Further research must be done to determine how to best apply the level streaming, and how to composite the world in a nice seamless manner.

[NOMEM-10] Asset finding and importing

Created: 10/Feb/16 Updated: 24/Feb/16 Resolved: 24/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Incomplete	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 hour		
Original Estimate:	Not Specified		

Sprint:	Setup Sprint, A New Hope
Story Points:	3

Description

Find models, sound, music, etc. for use later.

Comments

Comment by Per-Arne Waaler Stenshagen [17/Feb/16]

Laptop issues, pushed onwards to next sprint.

Comment by Bendik Hillestad [24/Feb/16]

Found some things, will import as needed.

[NOMEM-9] Draw basic maps to base on for the future

Created: 10/Feb/16 Updated: 17/Feb/16 Resolved: 17/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Minor
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	2 hours		
Original Estimate:	Not Specified		

Sprint:	Setup Sprint
Story Points:	1

Description

Draw a few maps (in paint or other) which future implementations may be based on.

Comments

Comment by Per-Arne Waaler Stenshagen [17/Feb/16]

Created some maps in Paint, easy to use as inspiration later.

[NOMEM-8] Settings Menu Layout Created: 10/Feb/16 Updated: 22/Feb/16 Resolved: 20/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day, 7 hours		
Original Estimate:	Not Specified		

Sprint:	Setup Sprint, A New Hope
Story Points:	3

Comments

Comment by Bjørn Nødland Fuglestad [20/Feb/16]

finished the layout for the settings menu. Does not have it functionality implemented yet. Will most likely do this in a later issue.

[NOMEM-7] Main Menu Branch Setup Created: 10/Feb/16 Updated: 16/Feb/16 Resolved: 16/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 day		
Original Estimate:	Not Specified		

Sprint:	Setup Sprint
Story Points:	3

Comments

Comment by Bjørn Nødland Fuglestad [16/Feb/16]
Finished the functionality of the main menu, but it needs some design to make it look nicer.

[NOMEM-6] Create sections and subsections for bachelor thesis

Created: 03/Feb/16 Updated: 05/Feb/16 Due: 10/Feb/16 Resolved: 05/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	1 hour		
Time Spent:	2 hours		
Original Estimate:	3 hours		
Environment:	Latex		

Sprint:	Initilizer sprint
Story Points:	1

Description

Fill in an estimate of all the sections and subsections we will need in the thesis.

Comments

Comment by Bjørn Nødland Fuglestad [05/Feb/16]

Created the sections and subsections that where common in other bachelor projects.

Will most likely change in the future.

[NOMEM-5] Contact Serapta Created: 03/Feb/16 Updated: 16/Feb/16 Resolved: 16/Feb/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	New Feature	**Priority:**	Critical
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	0 minutes		
Time Spent:	1 hour		
Original Estimate:	Not Specified		
Environment:	E-mail, Skype, Telephone, SMS, other		

Sprint:	Initilizer sprint, Setup Sprint
Story Points:	2

Description

Schedule a meeting with the Serapta representative.

Comments

Comment by Bjørn Nødland Fuglestad [10/Feb/16]
E-Mail was sent on the 8th of February.
Comment by Bjørn Nødland Fuglestad [15/Feb/16]
Received reply from email. Will meet at serapta on the 2 of march. Still need to set the time.
Comment by Bjørn Nødland Fuglestad [16/Feb/16]
Made an appointment with sarapta. Will be meeting 02.03.2016 at 12:00 am.

[NOMEM-4] Learn Latex coding Created: 03/Feb/16 Updated: 11/May/16 Due: 01/May/16 Resolved: 11/May/16	
Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Major
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Per-Arne Waaler Stenshagen
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		
Environment:	Latex (Shared latex)		

Sprint:	The Saga Ends
Story Points:	0

Description

Everyone will need to learn Latex sooner or later, preferably before the 1st of May.

[NOMEM-3] Work on the game design document

Created: 03/Feb/16 Updated: 10/Feb/16 Resolved: 10/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Improvement	**Priority:**	Blocker
Reporter:	Per-Arne Waaler Stenshagen	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	4 days, 3 hours		
Time Spent:	5 hours		
Original Estimate:	1 week		
Environment:	Google document		

Sprint:	Initilizer sprint
Story Points:	5

Description

Need to agree on something more wholesome in the design of the game, especially including game mechanics

Comments

Comment by Bjørn Nødland Fuglestad [05/Feb/16]

We are all working on this even though it is assigned to me.

Comment by Bjørn Nødland Fuglestad [10/Feb/16]

Decided enough to start coding in the next sprint. Will most likely be expanded as the game is worked on.

[NOMEM-2] Create index webpage for the project hig site.

Created: 01/Feb/16 Updated: 10/Feb/16 Due: 18/May/16 Resolved: 10/Feb/16

Status:	Resolved
Project:	NoMemoryProject
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Task	**Priority:**	Minor
Reporter:	Bjørn Nødland Fuglestad	**Assignee:**	Bjørn Nødland Fuglestad
Resolution:	Fixed	**Votes:**	0
Labels:	None		
Remaining Estimate:	2 days, 1 hour		
Time Spent:	Not Specified		
Original Estimate:	1 day		
Environment:	HTML		

Sprint:	Initilizer sprint
Story Points:	1

Description

Create the home page for the bachelor.

Comments

Comment by Bjørn Nødland Fuglestad [04/Feb/16]

Finished the layout need to fill in some content to the pages.

Comment by Bjørn Nødland Fuglestad [10/Feb/16]

The base for the webpage is done. Content will be added later.

Generated at Wed May 18 09:33:58 CEST 2016 by Per-Arne Waaler Stenshagen using JIRA 6.3.4#6332-sha1:51bc225ef474afe3128b2f66878477f322397b16.

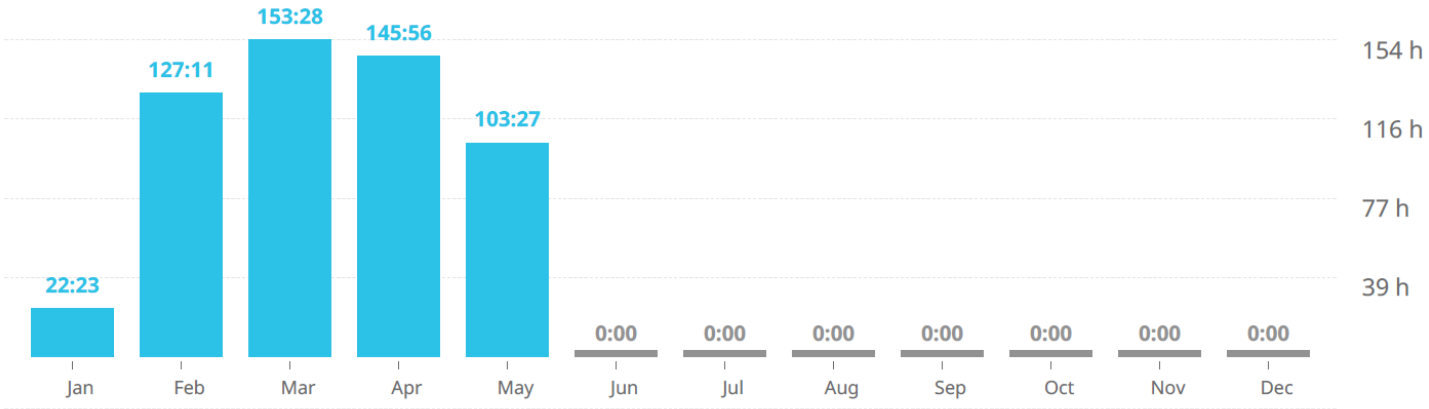
J Time Logs

These are the time logs for each individual member. First is Bendik's, second is Bjørn's and third is Per-Arne's. The first two are exported from Toggl, while the last one is exported from Microsoft Excel. Some time entries may be missing, but these logs should be mostly complete.

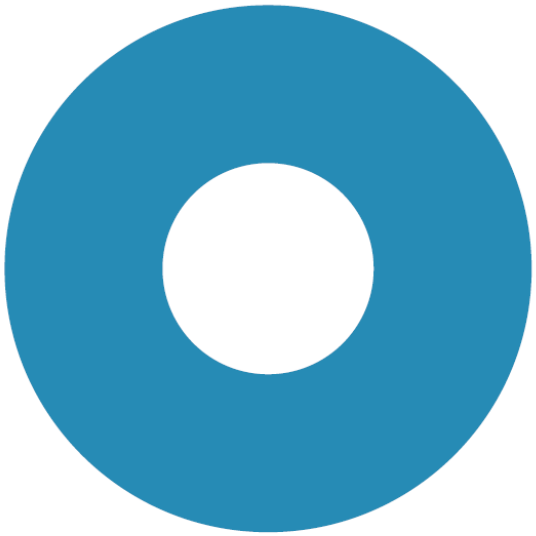
Summary report - Bendik

2016-01-01 - 2016-12-31
Total 552 h 27 min

Bachelor selected as projects

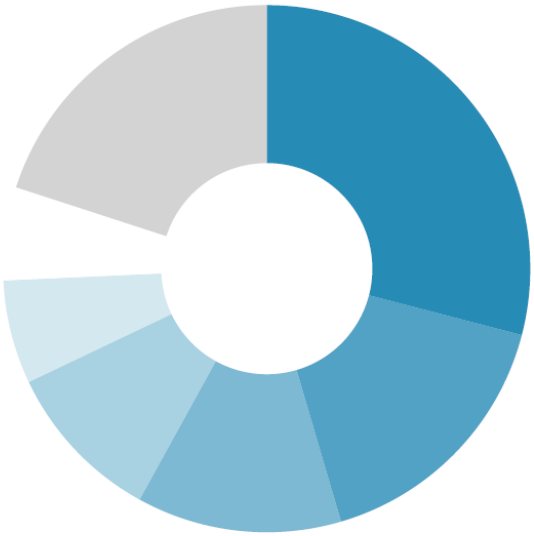


Projects



Bachelor 552:27:11

Time entries



Level 160:30:21
Research 90:35:46
Lighting 69:24:13
Bachelor report 54:45:58
Camera 35:09:12
Terrain 31:33:44
Other 110:27:57

Projects / Time entries - Bendik	Duration
Bachelor	**552:27:11**
Bachelor report	54:45:58
Bachelor thesis info	1:00:00
Camera	35:09:12
Campaign selection	7:48:14
Level	160:30:21
Level streaming	5:43:05
Level Streaming Research	15:14:04
Level streaming tests	14:59:35
Lighting	69:24:13
Research	90:35:46
Sarepta Meeting	5:00:00
Sprint meeting	14:52:30
Sprint meeting & Bachelor report	11:18:01
Terrain	31:33:44
Various	25:46:46
Working on design document	3:05:41
Working on project plan	5:40:01

Created with toggl.com

Detailed report

2016-01-01 - 2016-12-31

Total 680 h 18 min

NoMemoryTime selected as projects

Date	Description	Duration	User
01-05	**Discussing game genres for the project.** NoMemoryTime - [Documenting, Meeting]	**0:45:00** 13:40-14:25	Bjorn Fuglestad
01-06	**Eliminating game genres.** NoMemoryTime - [Documenting, Meeting]	**0:37:00** 16:44-17:21	Bjorn Fuglestad
01-07	**Discussed project in group and with Simon.** NoMemoryTime - [Documenting, Meeting]	**2:45:00** 13:00-15:45	Bjorn Fuglestad
01-11	**Working on documenting** NoMemoryTime - [Documenting]	**3:00:00** 04:30-07:30	Bjorn Fuglestad
01-11	**Meeting discussing project tools and development method.** NoMemoryTime - [Documenting, Meeting]	**2:00:00** 12:15-14:15	Bjorn Fuglestad
01-13	**Group meeting deciding on game and on tools.** NoMemoryTime - [Documenting, Meeting]	**3:30:00** 00:15-03:45	Bjorn Fuglestad
01-13	**Setup latex and working on tutorials.** NoMemoryTime - [Documenting, Latex]	**3:00:00** 17:15-20:15	Bjorn Fuglestad
01-14	**Fixing latex preview.** NoMemoryTime - [Fixing, Latex]	**3:00:00** 12:00-15:00	Bjorn Fuglestad
01-14	**Writing documentation in latex.** NoMemoryTime - [Documenting, Latex]	**2:00:00** 16:00-18:00	Bjorn Fuglestad
01-15	**Fixed confluence and added all the meeting** NoMemoryTime - [Documenting, Fixing]	**3:00:00** 13:11-16:11	Bjorn Fuglestad
01-18	**Create project plan google doc.** NoMemoryTime - [Documenting]	**1:00:00** 11:00-12:00	Bjorn Fuglestad
01-18	**Added time usage to toggle** NoMemoryTime - [Documenting]	**0:32:00** 14:15-14:47	Bjorn Fuglestad
01-19	**Working on project plan.** NoMemoryTime - [Documenting, mobile]	**4:34:06** 12:28-17:02	Bjorn Fuglestad
01-20	**meeting** NoMemoryTime - [Documenting, Meeting, mobile]	**0:15:06** 12:52-13:07	Bjorn Fuglestad
01-25	**Documenting** NoMemoryTime - [Documenting, mobile]	**5:14:07** 11:19-16:33	Bjorn Fuglestad
01-26	**Project plan** NoMemoryTime - [Documenting, mobile]	**5:14:01** 12:05-17:19	Bjorn Fuglestad

01-27	**Working on project plan.** NoMemoryTime - [Documenting, Meeting, mobile]	**3:40:19** 12:21-16:01	Bjorn Fuglestad
01-28	**Finished project plan** NoMemoryTime - [Documenting, mobile]	**1:00:00** 02:00-03:00	Bjorn Fuglestad
02-01	**Adding decisions to confluence** NoMemoryTime - [Documenting]	**1:03:09** 12:21-13:24	Bjorn Fuglestad
02-01	**Creating pages in confluence** NoMemoryTime - [Documenting]	**0:46:06** 13:29-14:15	Bjorn Fuglestad
02-01	**Working on home page for the bachelor** NoMemoryTime - [Coding, Documenting]	**0:56:56** 14:15-15:12	Bjorn Fuglestad
02-02	**Working on sharelatex document for bachelor thisis.** NoMemoryTime - [Documenting]	**2:33:10** 13:40-16:13	Bjorn Fuglestad
02-03	**Sprint meeting.** NoMemoryTime - [Meeting, mobile]	**0:58:00** 12:55-13:53	Bjorn Fuglestad
02-04	**Finding a good html editor to use.** NoMemoryTime - [research]	**0:43:00** 13:47-14:30	Bjorn Fuglestad
02-04	**Working on home page for the bachelor** NoMemoryTime - [Coding]	**0:37:06** 14:32-15:09	Bjorn Fuglestad
02-04	**Working on home page for the bachelor** NoMemoryTime - [Coding]	**5:24:28** 15:19-20:44	Bjorn Fuglestad
02-05	**Adding sections to thesis.** NoMemoryTime - [Documenting, Latex]	**1:49:21**	Bjorn Fuglestad
02-05	**Game design.** NoMemoryTime - [Meeting, mobile]	**2:05:00** 14:03-16:08	Bjorn Fuglestad
02-07	**Added a blank unreal project to the repository.** NoMemoryTime	**0:48:21** 13:31-14:19	Bjorn Fuglestad
02-10	**Sprint meeting.** NoMemoryTime - [Documenting, Meeting, mobile]	**2:41:13** 12:19-15:00	Bjorn Fuglestad
02-11	**Researching coding in unreal** NoMemoryTime - [research]	**3:04:28** 14:36-17:41	Bjorn Fuglestad
02-12	**Researching coding in unreal** NoMemoryTime - [research]	**4:12:36** 13:22-17:35	Bjorn Fuglestad
02-15	**Working on menu.** NoMemoryTime - [Coding, research]	**1:55:05** 11:01-12:56	Bjorn Fuglestad
02-15	**Working on menu.** NoMemoryTime - [Coding, research]	**1:19:06** 13:13-14:33	Bjorn Fuglestad
02-15	**Working on menu.** NoMemoryTime - [Coding, research]	**4:29:12** 15:47-20:17	Bjorn Fuglestad
02-16	**Working on menu.** NoMemoryTime - [Coding, research]	**2:00:00** 12:00-14:00	Bjorn Fuglestad

02-16	**Finished working on main menu.** NoMemoryTime - [Coding, research]	**7:15:13** 13:59-21:14	Bjorn Fuglestad
02-17	**Finishing setup sprint** NoMemoryTime - [Meeting]	**1:28:00** 12:06-13:34	Bjorn Fuglestad
02-17	**Talking with Mariusz.** NoMemoryTime - [Meeting]	**0:22:01**	Bjorn Fuglestad
02-17	**Setting up next sprint.** NoMemoryTime - [Documenting, Meeting, mobile]	**0:30:10** 13:46-14:16	Bjorn Fuglestad
02-17	**Finished working on Settings Menu.** NoMemoryTime - [Coding]	**2:31:35** 15:35-18:06	Bjorn Fuglestad
02-17	**Working in the sound settings.** NoMemoryTime - [Coding, research]	**2:17:44** 18:44-21:01	Bjorn Fuglestad
02-18	**Creating layout for other menus** NoMemoryTime - [Coding]	**1:18:00** 15:11-16:29	Bjorn Fuglestad
02-18	**Working on Graphics menu** NoMemoryTime - [Coding, research]	**2:06:52** 17:22-19:29	Bjorn Fuglestad
02-19	**Changing layout for main and settings menu.** NoMemoryTime - [Coding]	**2:51:00** 16:56-19:47	Bjorn Fuglestad
02-20	**Changed some of the layout and made keybinding menu.** NoMemoryTime - [Coding, research]	**1:52:04** 12:02-13:55	Bjorn Fuglestad
02-22	**Working on Instace menu** NoMemoryTime - [Coding, research]	**3:39:42** 11:28-15:08	Bjorn Fuglestad
02-22	**Creating layout for Campaign menu.** NoMemoryTime - [Coding]	**1:51:09** 15:27-17:18	Bjorn Fuglestad
02-22	**Creating layout for Conquest menu.** NoMemoryTime - [Coding]	**1:13:47** 17:18-18:32	Bjorn Fuglestad
02-23	**Creating layout for Conquest menu.** NoMemoryTime - [Coding]	**1:03:02** 12:55-13:58	Bjorn Fuglestad
02-23	**Resarching unreal.** NoMemoryTime - [research]	**2:08:25** 13:59-16:07	Bjorn Fuglestad
02-23	**Resarching unreal.** NoMemoryTime - [research]	**1:40:00** 16:26-18:06	Bjorn Fuglestad
02-24	**Sprint meeting** NoMemoryTime - [Meeting]	**2:00:00** 12:00-14:00	Bjorn Fuglestad
02-24	**Working on passing campaign iformation between levels.** NoMemoryTime - [Coding, research]	**5:14:16** 15:47-21:02	Bjorn Fuglestad
02-25	**Setting data in campaign menu and having it saved and loaded in campaign level** NoMemoryTime - [Coding, research]	**4:18:35** 17:10-21:29	Bjorn Fuglestad
02-26	**Closing old issue and opening new one.** NoMemoryTime - [Fixing]	**0:47:27** 17:14-18:02	Bjorn Fuglestad

02-26	**Source Tree Problem Trying to fix** NoMemoryTime	**0:19:00** 18:02-18:21	Bjorn Fuglestad
02-26	**Researched tiles in unreal and fixed source tree** NoMemoryTime - [Fixing, research]	**2:00:54** 18:25-20:26	Bjorn Fuglestad
02-27	**Researched tiles in unreal** NoMemoryTime - [research]	**0:58:18** 11:28-12:26	Bjorn Fuglestad
02-28	**Researching landscapes in unreal** NoMemoryTime - [research]	**1:55:33** 16:33-18:28	Bjorn Fuglestad
02-28	**Checking the free unreal engine assets** NoMemoryTime - [research]	**2:27:59** 18:50-21:18	Bjorn Fuglestad
02-29	**Filling in the introduction in the bachelor thesis** NoMemoryTime - [Coding, Documenting, Latex]	**3:44:01** 11:53-15:37	Bjorn Fuglestad
02-29	**Filling in the development process in the bachelor thesis** NoMemoryTime - [Coding, Documenting, Latex]	**3:50:08** 16:22-20:12	Bjorn Fuglestad
03-01	**Researched cameras in unreal** NoMemoryTime - [Coding, research]	**3:18:39** 10:08-13:27	Bjorn Fuglestad
03-01	**More research on cameras** NoMemoryTime - [Coding, research]	**4:31:38** 13:27-17:59	Bjorn Fuglestad
03-02	**Meeting serapta in Hamar** NoMemoryTime - [Meeting]	**5:00:00** 11:00-16:00	Bjorn Fuglestad
03-02	**Sprint meeting** NoMemoryTime - [Meeting]	**2:16:58** 16:30-18:47	Bjorn Fuglestad
03-03	**Creating UI the campaing** NoMemoryTime - [Coding]	**2:33:59** 09:31-12:05	Bjorn Fuglestad
03-03	**Finished setting up a campaign UI.** NoMemoryTime - [Coding, research]	**2:57:36** 12:19-15:17	Bjorn Fuglestad
03-03	**Added content to the project web page** NoMemoryTime - [Coding, Documenting]	**2:15:31** 15:17-17:33	Bjorn Fuglestad
03-04	**Implemented changing sound volume.** NoMemoryTime - [Coding, research]	**4:50:19** 09:26-14:17	Bjorn Fuglestad
03-04	**Working on moving units.** NoMemoryTime - [Coding, research]	**2:34:52** 14:22-16:57	Bjorn Fuglestad
03-05	**Tried moving topdown camera to c++** NoMemoryTime - [Coding]	**1:22:02** 11:26-12:48	Bjorn Fuglestad
03-05	**Working on moving units.** NoMemoryTime - [Coding, research]	**2:58:51** 12:48-15:47	Bjorn Fuglestad
03-05	**Created test implementation of selecting and mocing a unit.** NoMemoryTime - [Coding, research]	**3:00:54** 16:04-19:05	Bjorn Fuglestad
03-06	**Applied decal to selected units instead of text.** NoMemoryTime - [Coding, research]	**3:00:57** 11:58-14:59	Bjorn Fuglestad

03-06	**Added zoom to the camera** NoMemoryTime - [Coding, research]	**0:37:26** 14:59-15:37	Bjorn Fuglestad
03-06	**Worked on creating a grouped unit.** NoMemoryTime - [Coding, research]	**2:23:18** 15:37-18:00	Bjorn Fuglestad
03-07	**Made a simple attack system based on range** NoMemoryTime - [Coding, research]	**2:06:02** 10:29-12:35	Bjorn Fuglestad
03-07	**added decals for displaying different information.** NoMemoryTime - [Coding, research]	**1:00:08** 12:35-13:35	Bjorn Fuglestad
03-07	**Thinking about unit behavior** NoMemoryTime - [mobile, research]	**1:35:51** 13:35-15:11	Bjorn Fuglestad
03-07	**Working on turn based switching.** NoMemoryTime - [Coding, mobile, research]	**2:49:37** 15:21-18:11	Bjorn Fuglestad
03-08	**Fixing the animations** NoMemoryTime - [research]	**2:46:21** 10:12-12:59	Bjorn Fuglestad
03-08	**Transfered moving charecter from test project to main project** NoMemoryTime - [Coding]	**1:44:47** 13:10-14:55	Bjorn Fuglestad
03-08	**Worked on unit stats menu.** NoMemoryTime - [Coding]	**1:35:03** 15:36-17:11	Bjorn Fuglestad
03-09	**Lecture about writing Bachelor.** NoMemoryTime	**0:39:44**	Bjorn Fuglestad
03-09	**Sprint meeting** NoMemoryTime - [Meeting]	**2:09:49** 13:01-15:10	Bjorn Fuglestad
03-09	**Adding more options to unit controller.** NoMemoryTime - [Coding, research]	**4:12:37** 16:44-20:57	Bjorn Fuglestad
03-10	**Working on attacks and base unit** NoMemoryTime - [Coding, research]	**5:04:00** 09:10-14:14	Bjorn Fuglestad
03-10	**Working on attacks and base unit** NoMemoryTime - [Coding, research]	**2:51:41** 14:21-17:13	Bjorn Fuglestad
03-11	**Made unit attck.** NoMemoryTime - [Coding, research]	**3:27:28** 09:24-12:52	Bjorn Fuglestad
03-11	**Working on unit stat UI** NoMemoryTime - [Coding, research]	**4:26:33** 16:57-21:24	Bjorn Fuglestad
03-12	**Researching different Health bars and layouts** NoMemoryTime - [research]	**2:06:08** 09:42-11:48	Bjorn Fuglestad
03-12	**More pondering on layout** NoMemoryTime - [research]	**1:44:31** 12:22-14:06	Bjorn Fuglestad
03-12	**Change HealthLayout** NoMemoryTime - [Coding, research]	**0:37:00** 15:04-15:41	Bjorn Fuglestad
03-12	**Converted the base character into c++** NoMemoryTime - [Coding]	**0:26:02** 18:53-19:19	Bjorn Fuglestad

03-12	**Create base attack in c++** NoMemoryTime - [Coding]	**0:56:13** 19:19-20:15	Bjorn Fuglestad
03-12	**Removing redirectors** NoMemoryTime - [Fixing]	**0:17:34** 20:15-20:32	Bjorn Fuglestad
03-13	**Createde a simple attack bar** NoMemoryTime - [Coding, research]	**2:07:28** 11:17-13:24	Bjorn Fuglestad
03-13	**Changing the attack bar layout.** NoMemoryTime - [Coding]	**2:18:38** 13:28-15:47	Bjorn Fuglestad
03-13	**Converting blueprints to c++** NoMemoryTime - [Coding]	**1:25:47** 16:11-17:37	Bjorn Fuglestad
03-14	**Working on turn based switching.** NoMemoryTime - [Coding, research]	**3:33:12** 09:48-13:21	Bjorn Fuglestad
03-14	**Working on turn based switching.** NoMemoryTime - [Coding, research]	**2:36:42** 13:33-16:10	Bjorn Fuglestad
03-14	**Thinking and changing blueprint to c++** NoMemoryTime - [Coding, research]	**2:05:57** 16:35-18:41	Bjorn Fuglestad
03-14	**Moving more of character into c++** NoMemoryTime - [Coding, research]	**1:01:02** 18:53-19:54	Bjorn Fuglestad
03-15	**Added the rest of character into c++** NoMemoryTime - [Coding, research]	**2:04:17** 10:36-12:40	Bjorn Fuglestad
03-15	**Checking out other games for inspiration.** NoMemoryTime - [research]	**0:56:20** 14:15-15:11	Bjorn Fuglestad
03-15	**Added escape menu to skirmish** NoMemoryTime - [Coding]	**0:31:28** 17:46-18:17	Bjorn Fuglestad
03-16	**Prepare for meeting** NoMemoryTime - [Meeting]	**0:37:29** 09:26-10:04	Bjorn Fuglestad
03-16	**Sprint meeting** NoMemoryTime - [Meeting]	**1:45:00** 12:00-13:45	Bjorn Fuglestad
03-16	**Sending sprint reports** NoMemoryTime - [Documenting]	**1:31:17** 14:46-16:17	Bjorn Fuglestad
03-16	**Character values and leveling** NoMemoryTime - [Coding]	**3:06:22** 16:42-19:48	Bjorn Fuglestad
03-17	**Moved leveling and Selected to c++** NoMemoryTime - [Coding, research]	**1:57:37** 09:19-11:16	Bjorn Fuglestad
03-17	**Adding and thinking about character** NoMemoryTime - [Coding, research]	**2:07:57** 12:10-14:18	Bjorn Fuglestad
03-17	**Researching Character stats systems** NoMemoryTime - [Documenting, research]	**6:19:23** 15:43-22:02	Bjorn Fuglestad
03-18	**Researching Character stats systems** NoMemoryTime - [Documenting, research]	**4:17:06** 09:18-13:35	Bjorn Fuglestad

03-18	**Researching Character stats systems** NoMemoryTime - [Coding, Documenting, research]	**4:13:50** 14:11-18:25	Bjorn Fuglestad
03-19	**Researching Character stats systems** NoMemoryTime - [Coding, Documenting, research]	**2:42:21** 09:50-12:32	Bjorn Fuglestad
03-19	**Coding attribute system** NoMemoryTime - [Coding, Documenting]	**2:53:36** 13:49-16:43	Bjorn Fuglestad
03-19	**Adapting modefiers to c++** NoMemoryTime - [Coding, Documenting]	**3:32:34** 16:58-20:30	Bjorn Fuglestad
03-20	**Finished creating stat system in c++** NoMemoryTime - [Coding, Documenting]	**6:08:07** 10:26-16:34	Bjorn Fuglestad
03-20	**Worked on stat collectoin** NoMemoryTime - [Coding, Documenting]	**2:38:58** 17:25-20:04	Bjorn Fuglestad
03-21	**Finished moving main stat code into unreal project** NoMemoryTime - [Coding]	**3:55:06** 12:56-16:51	Bjorn Fuglestad
03-21	**Implemented statcollection in unreal** NoMemoryTime - [Coding]	**2:47:57** 17:44-20:31	Bjorn Fuglestad
03-22	**Added comments and UPropertys to the Stat system** NoMemoryTime - [Coding, Documenting]	**2:37:21** 10:28-13:05	Bjorn Fuglestad
03-22	**Fixed bugs in stat collection and amost finished the initializer** NoMemoryTime - [Coding, Debugging, Documenting, research]	**5:34:02** 15:43-21:17	Bjorn Fuglestad
03-23	**Started on constructor for stat collection** NoMemoryTime - [Coding]	**0:58:17** 11:04-12:02	Bjorn Fuglestad
03-23	**Adding constructors to all the stats** NoMemoryTime - [Coding, research]	**1:22:34** 12:04-13:26	Bjorn Fuglestad
03-23	**Moving structs and enums into own header** NoMemoryTime - [Coding]	**0:29:50** 13:26-13:56	Bjorn Fuglestad
03-23	**Documenting of new stat design** NoMemoryTime - [Documenting]	**0:54:45** 13:57-14:51	Bjorn Fuglestad
03-23	**Adding default constructors** NoMemoryTime - [Coding, research]	**0:29:04** 14:51-15:21	Bjorn Fuglestad
03-23	**Reading up un construction** NoMemoryTime - [Coding, Fixing, research]	**1:19:01** 15:27-16:46	Bjorn Fuglestad
03-23	**Finishing** NoMemoryTime - [Coding, research]	**1:22:00** 17:15-18:37	Bjorn Fuglestad
03-23	**Replacing stats in base character** NoMemoryTime - [Coding]	**1:11:40** 18:38-19:50	Bjorn Fuglestad
03-24	**Working on weapons and base character.** NoMemoryTime - [Coding, Debugging, research]	**5:10:37** 10:35-15:46	Bjorn Fuglestad

03-24	**Fixed the creaton of collection, more weapon work** NoMemoryTime - [Coding, research]	**2:22:32** 16:14-18:36	Bjorn Fuglestad
03-24	**Fixed the sword mesh** NoMemoryTime - [Coding, Fixing, research]	**0:55:13** 18:36-19:31	Bjorn Fuglestad
03-25	**Added stats modifers to weapons** NoMemoryTime - [Coding, research]	**2:42:07** 10:13-12:55	Bjorn Fuglestad
03-25	**Documenting of new weapon desing** NoMemoryTime - [Documenting]	**0:28:18** 12:55-13:24	Bjorn Fuglestad
03-25	**Apply modifiers to character on equip and restor AP on begin turn.** NoMemoryTime - [Coding]	**1:26:28** 13:24-14:50	Bjorn Fuglestad
03-25	**polishing character class** NoMemoryTime - [Coding]	**1:22:17** 15:14-16:36	Bjorn Fuglestad
03-25	**WOrking on selected decal** NoMemoryTime - [Coding]	**0:21:32** 18:34-18:55	Bjorn Fuglestad
03-26	**Implemented critical hit chance** NoMemoryTime - [Coding]	**0:11:21** 10:02-10:13	Bjorn Fuglestad
03-26	**Pushing and pulling new branch** NoMemoryTime	**0:10:57** 10:13-10:24	Bjorn Fuglestad
03-26	**Working on unit stat UI** NoMemoryTime - [Coding]	**2:50:18** 10:24-13:14	Bjorn Fuglestad
03-26	**Working on unit stat UI** NoMemoryTime - [Coding]	**0:48:00** 13:19-14:07	Bjorn Fuglestad
03-26	**Creating action bar** NoMemoryTime - [Coding]	**1:59:33** 15:01-17:01	Bjorn Fuglestad
03-28	**Improving UI** NoMemoryTime - [Coding, research]	**4:23:19** 10:45-15:08	Bjorn Fuglestad
03-28	**Weapon Stats UI** NoMemoryTime - [Coding]	**2:59:33** 15:43-18:43	Bjorn Fuglestad
03-29	**Finished displaying health and action point** NoMemoryTime - [Coding, research]	**0:58:51** 10:59-11:58	Bjorn Fuglestad
03-29	**Fixed displaying percentage modifers** NoMemoryTime - [Coding, Fixing]	**0:21:14** 11:58-12:19	Bjorn Fuglestad
03-29	**Disable ui and pause game on open menu** NoMemoryTime - [Coding]	**0:10:49** 12:19-12:30	Bjorn Fuglestad
03-29	**Fixed displaying percentage in unit stats** NoMemoryTime - [Coding, Fixing]	**0:12:51** 12:32-12:44	Bjorn Fuglestad
03-29	**Hide unit stat bar when no unit is selected.** NoMemoryTime - [Coding]	**0:37:52** 12:44-13:22	Bjorn Fuglestad
03-29	**Fixed add percentage modifier** NoMemoryTime - [Coding, Fixing]	**0:36:00** 13:13-13:49	Bjorn Fuglestad

03-29	**Pushing and pulling new branch** NoMemoryTime	**0:31:49** 13:50-14:22	Bjorn Fuglestad
03-29	**Setting up linkers in stat collection** NoMemoryTime - [Coding]	**1:38:37** 14:22-16:01	Bjorn Fuglestad
03-29	**Merging into dev branch.** NoMemoryTime - [Documenting]	**0:21:51** 17:02-17:24	Bjorn Fuglestad
03-30	**Sprint meeting** NoMemoryTime - [Documenting, Meeting]	**2:37:00** 12:00-14:37	Bjorn Fuglestad
03-30	**Get new branch for implementing AP cost** NoMemoryTime	**0:03:27** 15:19-15:23	Bjorn Fuglestad
03-30	**Working on movement AP cost** NoMemoryTime	**3:55:18** 16:14-20:10	Bjorn Fuglestad
03-31	**Working on movement AP cost** NoMemoryTime - [Coding, research]	**5:10:59** 10:37-15:48	Bjorn Fuglestad
03-31	**Sending sprint reports** NoMemoryTime - [Documenting]	**0:17:54** 17:25-17:43	Bjorn Fuglestad
03-31	**Reseaching hit chance** NoMemoryTime - [research]	**0:38:20** 17:43-18:22	Bjorn Fuglestad
04-01	**Working on hit chance** NoMemoryTime - [Coding, research]	**4:18:25** 10:19-14:38	Bjorn Fuglestad
04-01	**Finished working on hit chance** NoMemoryTime - [Coding]	**1:41:50** 15:21-17:03	Bjorn Fuglestad
04-01	**Working on displaying info under mouse** NoMemoryTime - [Coding, research]	**2:34:15** 18:13-20:48	Bjorn Fuglestad
04-02	**Working on displaying info under mouse** NoMemoryTime - [Coding]	**6:27:20** 11:48-18:15	Bjorn Fuglestad
04-02	**Working on displaying info under mouse** NoMemoryTime - [Coding]	**0:43:24** 18:48-19:31	Bjorn Fuglestad
04-03	**Created a basic floating text on damage** NoMemoryTime - [Coding, research]	**3:00:53** 12:06-15:06	Bjorn Fuglestad
04-03	**Working on navigation spline** NoMemoryTime - [Coding, research]	**0:56:18** 15:07-16:03	Bjorn Fuglestad
04-03	**Working on navigation spline** NoMemoryTime - [Coding, research]	**2:25:04** 16:35-19:00	Bjorn Fuglestad
04-04	**Created hover over health** NoMemoryTime - [Coding]	**1:11:16** 10:45-11:56	Bjorn Fuglestad
04-04	**Almost finished spline** NoMemoryTime - [Coding, research]	**2:53:17** 12:30-15:23	Bjorn Fuglestad
04-04	**Fixed units not moving around each other** NoMemoryTime - [Coding, Fixing, research]	**0:49:36** 16:39-17:28	Bjorn Fuglestad

04-04	**Fixed updating navmesh** NoMemoryTime - [Coding, Fixing, research]	**1:45:21** 17:28-19:13	Bjorn Fuglestad
04-05	**Minor fixes for displaying info** NoMemoryTime - [Coding, Fixing, research]	**1:17:11** 11:00-12:18	Bjorn Fuglestad
04-05	**Pushing and pulling new branch** NoMemoryTime - [Documenting]	**0:19:29** 12:18-12:37	Bjorn Fuglestad
04-05	**Made simple volume spawner** NoMemoryTime - [Coding, research]	**1:30:00** 12:38-14:08	Bjorn Fuglestad
04-05	**Changed unit init and spawning from array.** NoMemoryTime - [Coding, Fixing, research]	**1:28:14** 15:20-16:48	Bjorn Fuglestad
04-05	**Fixing some details** NoMemoryTime - [Coding, Fixing]	**0:12:22** 16:48-17:00	Bjorn Fuglestad
04-06	**Sprint meeting** NoMemoryTime - [Documenting, Meeting]	**1:23:00** 12:10-13:33	Bjorn Fuglestad
04-06	**Documenting meeting and sending meeting raports.** NoMemoryTime - [Documenting]	**0:41:46** 14:24-15:05	Bjorn Fuglestad
04-06	**Improving Unit Spawner** NoMemoryTime - [Coding, research]	**2:28:32** 16:07-18:35	Bjorn Fuglestad
04-06	**Finished unit spawner with pattern.** NoMemoryTime - [Coding]	**1:01:40** 19:05-20:06	Bjorn Fuglestad
04-07	**Filling in stats for new units** NoMemoryTime - [Coding]	**3:03:47** 10:47-13:50	Bjorn Fuglestad
04-07	**Filled in base stats for units** NoMemoryTime - [Coding]	**0:22:48** 14:00-14:23	Bjorn Fuglestad
04-07	**Checking out how to retarget animations** NoMemoryTime - [research]	**0:39:43** 14:25-15:05	Bjorn Fuglestad
04-07	**Created a bunch of weapons** NoMemoryTime - [Coding, research]	**2:15:08** 15:25-17:41	Bjorn Fuglestad
04-07	**Pushing infinity blade assets.** NoMemoryTime	**0:40:42** 17:43-18:24	Bjorn Fuglestad
04-07	**Working on characters for units and mosters** NoMemoryTime - [Coding, research]	**1:15:38** 18:39-19:54	Bjorn Fuglestad
04-08	**Creating moster blueprints** NoMemoryTime - [Coding]	**1:09:57** 10:48-11:58	Bjorn Fuglestad
04-08	**Replying to sprint emails** NoMemoryTime - [Documenting]	**0:33:00** 12:00-12:33	Bjorn Fuglestad
04-08	**Fixing collisions and appling stats** NoMemoryTime - [Coding]	**0:58:00** 13:00-13:58	Bjorn Fuglestad
04-08	**Creating animations** NoMemoryTime - [Coding, research]	**1:39:11** 13:58-15:37	Bjorn Fuglestad

04-08	**Troll, spider and grunt move animations**	**1:15:53**	Bjorn Fuglestad
	NoMemoryTime - [Coding, research]	15:55-17:11	
04-08	**Created monster weapons**	**0:59:17**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	17:16-18:15	
04-08	**Creating grunt master**	**1:21:01**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	18:52-20:13	
04-08	**Finished off units and weapons**	**0:37:55**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	20:13-20:51	
04-08	**Pushing and sending follow up emails.**	**0:23:58**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	20:51-21:15	
04-09	**working on fixing decals on character**	**2:42:18**	Bjorn Fuglestad
	NoMemoryTime - [Coding, Fixing]	10:40-13:22	
04-09	**Finish fixing decals**	**1:53:00**	Bjorn Fuglestad
	NoMemoryTime - [Coding, Fixing]	13:26-15:19	
04-09	**Started converting skirmish player to c++**	**1:05:33**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	16:00-17:05	
04-09	**Converting Skirmish player to c++**	**3:02:57**	Bjorn Fuglestad
	NoMemoryTime - [Coding, research]	18:12-21:15	
04-10	**Skirmish player actions in c++**	**0:29:06**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	10:53-11:22	
04-10	**Converting Skirmish player to c++**	**3:06:24**	Bjorn Fuglestad
	NoMemoryTime - [Coding, research]	11:41-14:48	
04-10	**Converting Skirmish player to c++**	**2:15:11**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	16:00-18:15	
04-10	**Converting Skirmish player to c++**	**3:11:16**	Bjorn Fuglestad
	NoMemoryTime - [Coding, research]	18:47-21:59	
04-11	**Finishing converiting skirmish player to c++**	**2:26:48**	Bjorn Fuglestad
	NoMemoryTime - [Coding, Fixing]	11:14-13:41	
04-11	**Working on skirmish game mode**	**3:01:33**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	14:22-17:24	
04-11	**Fixing hover over unit, fix popup message**	**0:58:07**	Bjorn Fuglestad
	NoMemoryTime - [Coding, Fixing]	17:41-18:39	
04-11	**Noting bugs and things that needs to be fixed**	**0:05:37**	Bjorn Fuglestad
	NoMemoryTime	18:40-18:46	
04-12	**Working on setting up skrimish UI**	**1:59:52**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	12:55-14:55	
04-12	**Creating victory condition in skimish game mode.**	**3:41:39**	Bjorn Fuglestad
	NoMemoryTime - [Coding, research]	15:40-19:22	
04-12	**Implementing skirmish game mode in c++**	**4:07:12**	Bjorn Fuglestad
	NoMemoryTime - [Coding]	19:50-23:57	

04-13	**Working on victory condition** NoMemoryTime - [Coding]	**2:38:39** 00:20-02:58	Bjorn Fuglestad
04-13	**Pulling to laptop** NoMemoryTime	**0:31:00** 03:00-03:31	Bjorn Fuglestad
04-13	**Sprint meeting** NoMemoryTime - [Meeting]	**2:36:00** 12:00-14:36	Bjorn Fuglestad
04-13	**Got victory condition working. Need to remove old code** NoMemoryTime - [Coding, Debugging, Fixing]	**1:54:40** 17:11-19:06	Bjorn Fuglestad
04-13	**Sending sprint reports** NoMemoryTime - [Documenting]	**0:49:20** 19:06-19:55	Bjorn Fuglestad
04-13	**Cleared out old code in skirmish game mode** NoMemoryTime - [Coding, Fixing]	**0:18:45** 20:23-20:42	Bjorn Fuglestad
04-14	**Added end game screen when winning or lossing** NoMemoryTime - [Coding]	**1:36:02** 11:40-13:16	Bjorn Fuglestad
04-14	**Preparing the demo** NoMemoryTime - [Coding, Fixing]	**2:22:17** 13:55-16:17	Bjorn Fuglestad
04-14	**Creating packaged project** NoMemoryTime - [Coding]	**3:44:42** 17:19-21:03	Bjorn Fuglestad
04-15	**Sending demo emails** NoMemoryTime - [Documenting]	**0:20:00** 16:50-17:10	Bjorn Fuglestad
04-16	**Creating load screen** NoMemoryTime - [Coding, research]	**3:02:41** 10:43-13:46	Bjorn Fuglestad
04-16	**Change git ignore, pull anim branch, look for animations** NoMemoryTime	**1:25:12** 13:46-15:11	Bjorn Fuglestad
04-16	**Retargeting skeleton for animation** NoMemoryTime	**0:27:12** 15:11-15:38	Bjorn Fuglestad
04-16	**Creating animation blueprint for humans** NoMemoryTime	**0:33:39** 15:38-16:12	Bjorn Fuglestad
04-17	**Setting up transitions for all units** NoMemoryTime - [Coding]	**3:09:08** 13:25-16:34	Bjorn Fuglestad
04-17	**Finished Adding animations to units** NoMemoryTime - [Coding, Fixing]	**0:58:35** 16:58-17:56	Bjorn Fuglestad
04-18	**Resarching damage types and events.** NoMemoryTime - [research]	**2:36:55** 12:47-15:24	Bjorn Fuglestad
04-18	**Implementing damage types** NoMemoryTime - [Coding]	**1:31:41** 16:12-17:44	Bjorn Fuglestad
04-19	**Implementing damage types** NoMemoryTime - [Coding, research]	**0:55:13** 11:09-12:05	Bjorn Fuglestad
04-19	**Researching damage types** NoMemoryTime - [research]	**2:07:35** 12:43-14:51	Bjorn Fuglestad

04-19	**Started implementing damageType again** NoMemoryTime - [Coding]	**0:29:20** 14:59-15:28	Bjorn Fuglestad
04-19	**Finished implementing Damage Types** NoMemoryTime - [Coding]	**4:09:56** 15:58-20:08	Bjorn Fuglestad
04-19	**Push and merge branch** NoMemoryTime	**0:11:12** 20:09-20:20	Bjorn Fuglestad
04-20	**Sprint meeting** NoMemoryTime - [Meeting]	**1:34:00** 12:02-13:36	Bjorn Fuglestad
04-20	**Sending sprint meeting log.** NoMemoryTime - [Documenting]	**0:50:59** 15:34-16:25	Bjorn Fuglestad
04-20	**Adding meeting notes to Confluence** NoMemoryTime - [Documenting]	**0:03:43** 16:25-16:29	Bjorn Fuglestad
04-20	**Removing starter content from repo** NoMemoryTime - [Fixing]	**0:16:41** 16:29-16:46	Bjorn Fuglestad
04-20	**Made attacker rotate towards defender** NoMemoryTime - [Coding, research]	**1:01:34** 20:38-21:39	Bjorn Fuglestad
04-21	**Implementing combat log** NoMemoryTime - [Coding, research]	**3:28:41** 11:13-14:41	Bjorn Fuglestad
04-21	**Adding line trace to check range and hit** NoMemoryTime - [Coding, research]	**1:57:35** 16:07-18:05	Bjorn Fuglestad
04-21	**Finished line trace for checking hit** NoMemoryTime - [Coding, research]	**2:37:52** 19:02-21:40	Bjorn Fuglestad
04-22	**Thinking about ranged attacks** NoMemoryTime - [research]	**0:38:39** 11:02-11:41	Bjorn Fuglestad
04-22	**Planning implementation of projectile** NoMemoryTime - [research]	**0:39:00** 14:20-14:59	Bjorn Fuglestad
04-22	**Almost done with making fireballs** NoMemoryTime - [Coding, research]	**6:04:46** 15:00-21:04	Bjorn Fuglestad
04-23	**Finished fire ball** NoMemoryTime - [Coding]	**1:50:22** 12:42-14:32	Bjorn Fuglestad
04-23	**Gave the trolls a rock to throw** NoMemoryTime - [Coding]	**0:35:22** 14:32-15:08	Bjorn Fuglestad
04-23	**Gathered data using profiler and fps chart** NoMemoryTime - [Documenting, research]	**1:39:35** 15:08-16:47	Bjorn Fuglestad
04-24	**Searched for some music, added some new issues** NoMemoryTime - [Documenting, research]	**0:50:26** 12:28-13:19	Bjorn Fuglestad
04-25	**Working on bachelor thisis** NoMemoryTime - [Documenting, mobile]	**4:07:58**	Bjorn Fuglestad
04-25	**Working on bachelor thisis** NoMemoryTime - [Documenting]	**1:58:08** 16:26-18:24	Bjorn Fuglestad

04-26	**Working on bachelor thisis** NoMemoryTime - [Documenting, Latex]	**2:39:19** 11:45-14:25	Bjorn Fuglestad
04-26	**Writing documentation** NoMemoryTime - [Documenting, Latex]	**1:33:00** 15:15-16:48	Bjorn Fuglestad
04-27	**Sprint meeting** NoMemoryTime - [Meeting, mobile]	**1:31:44**	Bjorn Fuglestad
04-27	**Sending sprint reports** NoMemoryTime - [Documenting]	**0:25:25** 15:11-15:36	Bjorn Fuglestad
04-27	**Looking for ways to clean repo** NoMemoryTime - [research]	**0:29:00** 17:03-17:32	Bjorn Fuglestad
04-27	**Looking for ways to clean repo** NoMemoryTime - [research]	**1:22:33** 19:15-20:38	Bjorn Fuglestad
04-28	**Making improvements to game based on feedback** NoMemoryTime - [Coding, Fixing]	**4:15:02** 12:47-17:02	Bjorn Fuglestad
04-28	**Making improvements to game based on feedback** NoMemoryTime - [Coding, Fixing]	**2:11:42** 17:39-19:50	Bjorn Fuglestad
04-29	**Making improvements to game based on feedback** NoMemoryTime - [Coding, Fixing, research]	**1:41:07** 11:19-13:00	Bjorn Fuglestad
04-29	**Adding starter content to google drive** NoMemoryTime - [Documenting, Fixing]	**0:12:57** 16:59-17:12	Bjorn Fuglestad
04-29	**Working on implementing outlines** NoMemoryTime - [Coding, research]	**2:06:29** 17:13-19:19	Bjorn Fuglestad
04-30	**Finished outlines with different colors** NoMemoryTime - [Coding, research]	**2:32:40** 10:59-13:31	Bjorn Fuglestad
04-30	**Better turn notification** NoMemoryTime - [Coding, Fixing]	**0:26:42** 13:40-14:07	Bjorn Fuglestad
04-30	**Adding sprint meeting notes from last sprint** NoMemoryTime - [Documenting]	**0:15:25** 14:09-14:24	Bjorn Fuglestad
04-30	**Updated webpage with new progress** NoMemoryTime - [Coding, Documenting]	**1:09:12** 14:24-15:33	Bjorn Fuglestad
04-30	**Commiting changes** NoMemoryTime - [Documenting]	**0:07:08** 15:40-15:47	Bjorn Fuglestad
04-30	**Looking into linetrace by object** NoMemoryTime - [research]	**0:30:05** 16:37-17:07	Bjorn Fuglestad
05-02	**Working on bachelor thisis** NoMemoryTime - [Documenting, Latex]	**5:35:00** 10:00-15:35	Bjorn Fuglestad
05-02	**Merging branches** NoMemoryTime - [Fixing]	**0:47:00** 17:00-17:47	Bjorn Fuglestad
05-02	**Uploadung images, doing performance analysis on campaign map** NoMemoryTime - [Documenting]	**0:41:35** 17:47-18:29	Bjorn Fuglestad

05-03	**Working on bachelor thisis** NoMemoryTime - [Documenting, Latex]	**5:34:00** 10:00-15:34	Bjorn Fuglestad
05-03	**Balancing the units** NoMemoryTime - [Coding, Documenting]	**0:58:41** 16:21-17:20	Bjorn Fuglestad
05-04	**Sprint meeting and writing bachelor** NoMemoryTime - [Documenting, Latex, Meeting, mobile]	**6:09:13** 10:03-16:12	Bjorn Fuglestad
05-05	**working on bachelor thisis** NoMemoryTime - [Documenting, Latex, mobile]	**5:33:05**	Bjorn Fuglestad
05-05	**Sending sprint reports** NoMemoryTime	**0:11:06** 16:17-16:28	Bjorn Fuglestad
05-06	**working on bachelor thisis** NoMemoryTime - [Documenting, Latex, mobile]	**5:21:03**	Bjorn Fuglestad
05-06	**working on bachelor thisis** NoMemoryTime - [Documenting, Latex, mobile]	**0:33:56**	Bjorn Fuglestad
05-07	**Balancing the units** NoMemoryTime - [Documenting, Fixing]	**3:57:20** 10:54-14:51	Bjorn Fuglestad
05-07	**Balancing the units** NoMemoryTime - [Documenting, Fixing]	**2:07:13** 15:08-17:15	Bjorn Fuglestad
05-07	**Preparing next branch** NoMemoryTime	**0:06:12** 17:15-17:21	Bjorn Fuglestad
05-07	**Adding damage range to display** NoMemoryTime - [Coding, Fixing]	**0:30:00** 17:49-18:19	Bjorn Fuglestad
05-07	**Adding damage range to display** NoMemoryTime - [Coding, Fixing]	**2:14:18** 18:20-20:34	Bjorn Fuglestad
05-08	**Loadscreens on return to menu** NoMemoryTime - [Coding, Fixing]	**0:44:42** 11:47-12:32	Bjorn Fuglestad
05-08	**Improving hover over text and over unit text.** NoMemoryTime - [Coding, Fixing]	**0:34:47** 12:32-13:06	Bjorn Fuglestad
05-08	**Fixed displaying modifiers** NoMemoryTime - [Coding, Fixing]	**0:58:20** 13:06-14:05	Bjorn Fuglestad
05-08	**Created ranged base class** NoMemoryTime - [Coding]	**1:02:43** 14:05-15:08	Bjorn Fuglestad
05-08	**Decorated Fields map** NoMemoryTime	**2:21:13** 15:09-17:30	Bjorn Fuglestad
05-08	**Changing brittle bridge map** NoMemoryTime - [Fixing]	**0:55:28** 17:30-18:26	Bjorn Fuglestad
05-08	**Creating bridge in map** NoMemoryTime	**0:26:00** 18:40-19:06	Bjorn Fuglestad
05-08	**Finishing bridges** NoMemoryTime	**0:56:10** 19:06-20:02	Bjorn Fuglestad

05-08	**Getting level images** NoMemoryTime	**0:10:00** 22:47-22:57	Bjorn Fuglestad
05-09	**Working on bachelor thisis** NoMemoryTime - [Documenting, Latex, mobile]	**4:57:12**	Bjorn Fuglestad
05-09	**Changing difficulty, creating demo and sending.** NoMemoryTime - [Coding, Fixing]	**4:37:28** 16:40-21:18	Bjorn Fuglestad
05-10	**Working on bachelor thisis** NoMemoryTime - [Documenting, Latex]	**5:00:00** 10:00-15:00	Bjorn Fuglestad
05-11	**Sprint meeting and writing bachelor** NoMemoryTime - [Documenting, Meeting, mobile]	**5:16:32** 09:59-15:15	Bjorn Fuglestad
05-11	**Tried some improvements, Added level description and image.** NoMemoryTime - [Coding, Fixing]	**1:16:13** 17:07-18:24	Bjorn Fuglestad
05-11	**Generated statistic files** NoMemoryTime - [Documenting]	**0:21:08** 18:24-18:45	Bjorn Fuglestad
05-11	**Not drawing navigation spline when within range** NoMemoryTime - [Coding, Fixing]	**0:31:09** 18:45-19:16	Bjorn Fuglestad
05-11	**Created health pickups** NoMemoryTime - [Coding]	**1:48:49** 19:16-21:05	Bjorn Fuglestad
05-11	**Finished making a weapon pickup** NoMemoryTime - [Coding]	**0:23:21** 21:11-21:34	Bjorn Fuglestad
05-11	**Pushing and getting new branch** NoMemoryTime	**0:08:00** 21:34-21:42	Bjorn Fuglestad
05-12	**Working on bachelor thisis** NoMemoryTime - [Documenting, Latex]	**5:10:08** 10:04-15:14	Bjorn Fuglestad
05-12	**Looking into making tutorial video.** NoMemoryTime - [Documenting, research]	**0:42:07** 16:37-17:20	Bjorn Fuglestad
05-12	**Sending sprint reports** NoMemoryTime - [Documenting]	**0:12:34** 17:20-17:32	Bjorn Fuglestad
05-13	**Working on bachelor thisis.** NoMemoryTime - [Documenting, Latex, mobile]	**5:49:39** 10:00-15:49	Bjorn Fuglestad
05-13	**Working on cutscene tutorial** NoMemoryTime - [Coding, Fixing, research]	**1:40:36** 16:49-18:30	Bjorn Fuglestad
05-14	**Working on cutscene tutorial** NoMemoryTime - [research]	**1:32:35** 10:06-11:39	Bjorn Fuglestad
05-14	**Looking into sound effects** NoMemoryTime - [research]	**0:13:28** 11:39-11:52	Bjorn Fuglestad
05-14	**Looking into sound effects** NoMemoryTime - [research]	**0:33:28** 12:21-12:55	Bjorn Fuglestad
05-14	**Looking into sound effects** NoMemoryTime - [research]	**1:53:12** 13:44-15:37	Bjorn Fuglestad

05-14	**Cleaning code** NoMemoryTime - [Coding, Fixing, Reviewing code]	**0:18:04** 15:37-15:55	Bjorn Fuglestad
05-14	**Cleaning code** NoMemoryTime - [Coding, Fixing, Reviewing code]	**1:06:36** 16:41-17:48	Bjorn Fuglestad
05-14	**Cleaning code** NoMemoryTime - [Coding, Fixing, Reviewing code]	**1:30:07** 18:24-19:54	Bjorn Fuglestad
05-15	**Cleaning code** NoMemoryTime - [Coding, Fixing, Reviewing code]	**5:18:12** 09:15-14:33	Bjorn Fuglestad
05-15	**Cleaning code** NoMemoryTime - [Coding, Fixing, Reviewing code]	**2:01:43** 15:18-17:20	Bjorn Fuglestad
05-15	**Cleaning code and making UML diagram for Stats** NoMemoryTime - [Coding, Documenting, Fixing]	**2:24:35** 17:50-20:15	Bjorn Fuglestad
05-15	**Getting ai branch** NoMemoryTime	**0:30:40** 20:15-20:45	Bjorn Fuglestad
05-15	**Looking into added AI** NoMemoryTime - [Reviewing code]	**0:14:28** 20:54-21:09	Bjorn Fuglestad
05-16	**Working on bachelor thisis.** NoMemoryTime - [Documenting, Latex]	**0:24:00** 09:44-10:08	Bjorn Fuglestad
05-16	**Working on bachelor thisis.** NoMemoryTime - [Documenting, Latex]	**5:31:03** 10:09-15:40	Bjorn Fuglestad
05-16	**Working on bachelor thisis.** NoMemoryTime - [Documenting, Latex]	**1:11:49** 16:50-18:02	Bjorn Fuglestad
05-16	**Working on bachelor thisis** NoMemoryTime - [Documenting, Latex]	**2:02:01** 18:02-20:04	Bjorn Fuglestad
05-16	**Working on bachelor thisis.** NoMemoryTime - [Documenting, Latex]	**3:11:40** 20:33-23:44	Bjorn Fuglestad
05-17	**Working on bachelor thisis.** NoMemoryTime - [Documenting, Latex]	**7:54:00** 09:48-17:42	Bjorn Fuglestad
05-17	**Finished making a tutorial** NoMemoryTime - [Coding, Fixing]	**1:49:19** 18:42-20:31	Bjorn Fuglestad
05-17	**working on bachelor thisis** NoMemoryTime - [Documenting, Latex]	**1:20:10** 20:31-21:51	Bjorn Fuglestad
05-17	**Preparing for packing** NoMemoryTime - [Coding, Documenting, Fixing]	**2:05:35** 22:19-00:25	Bjorn Fuglestad
05-18	**Creating game play videos.** NoMemoryTime - [Documenting]	**0:48:26** 00:25-01:13	Bjorn Fuglestad
05-18	**Finishing Bachelor** NoMemoryTime	**4:36:00** 07:24-12:00	Bjorn Fuglestad

Per-Arne Waaler Stenshagen

	h	min	
05.01.2016	1	45	
06.01.2016	1	30	
07.01.2016	2	45	
11.01.2016	2	15	
13.01.2016	3	30	
27.01.2016	4	30	

Initializer Sprint

03.02.2016	1	15	Sprint meeting
05.02.2016	2	5	

Setup sprint

10.02.2016	3	0	Sprint meeting
10.02.2016	1	15	
11.02.2016	3	15	
12.02.2016	2	45	
15.02.2016	4	30	
16.02.2016	5	0	
17.02.2016	4	30	

A new hope

17.02.2016	2	30	Sprint meeting
22.02.2016	7	15	
23.02.2016	6	45	

Empire strikes back

24.02.2016	2	0	Sprint meeting
24.02.2016	2	15	
25.02.2016	4	45	
26.02.2016	5	0	
29.02.2016	4	45	
01.02.2016	7	15	
02.03.2016	5	0	Sarepta meeting

Return of the Jedi

02.03.2016	1	0	Sprint meeting
03.03.2016	2	45	
04.03.2016	6	30	
06.03.2016	5	45	
07.03.2016	7	0	
08.03.2016	4	30	

Phantom Menace

09.03.2016	3	0	Sprint meeting
09.03.2016	1	15	
10.03.2016	6	30	
11.03.2016	4	15	
14.03.2016	6	15	
15.03.2016	4	15	

Attack of the Clones

16.03.2016	1	45	Sprint meeting

16.03.2016	2	0	
17.03.2016	3	30	
18.03.2016	2	30	
19.03.2016	2	0	
20.03.2016	3	15	
21.03.2016	2	45	
22.03.2016	2	30	
23.03.2016	4	0	
29.03.2016	5	15	

Revenge of the Sith

30.03.2016	1	25	Sprint meeting
30.03.2016	1	30	
31.03.2016	4	15	
01.04.2016	5	30	
02.04.2016	4	15	
03.04.2016	5	30	
04.04.2016	6	15	
05.04.2016	7	15	

The Force Awakens

06.04.2016	1	15	Sprint meeting
06.04.2016	1	15	
07.04.2016	4	0	
08.04.2016	5	45	
11.04.2016	6	15	

Knights of the old Repuplic

13.04.2016	2	30	Sprint meeting
13.04.2016	2	15	
14.04.2016	5	15	
15.04.2016	5	40	
16.03.2016	6	30	
17.04.2016	5	15	
18.04.2016	7	0	

Repuplic Commando

20.04.2016	2	30	Sprint meeting
21.04.2016	3	45	
22.04.2016	4	45	
23.04.2016	4	30	
25.04.2016	4	30	
26.04.2016	8	15	

Empire at War

27.04.2016	1	30	Sprint meeting
27.04.2016	3	0	
28.04.2016	6	15	
29.04.2016	8	0	
30.04.2016	9	15	
01.05.2016	9	15	
02.05.2016	6	0	
03.05.2016	6	0	

Rogue One

04.05.2016	5	30	Integrated sprint meeting
05.05.2016	7	0	
06.05.2016	9	0	
07.05.2016	7	45	
08.05.2016	9	30	
09.05.2016	7	15	
10.05.2016	8	15	

The Saga Ends

11.05.2016	5	15	Integrated sprint meeting
12.05.2016	7	0	
13.05.2016	9	30	
14.05.2016	8	30	
15.05.2016	13	15	
16.05.2016	11	40	
17.05.2016	9	45	
Total Sum:	459	h	

K Performance Charts

The following pages contain performance charts for specific levels created using Microsoft Excel 2010©, and exported as PDF files. The data found in these charts have been created on a desktop computer with the specifications detailed in Table 1. This data has also been used to create Figure 47, showing the average amount of time spent in each phase of creating the frames for the different levels:

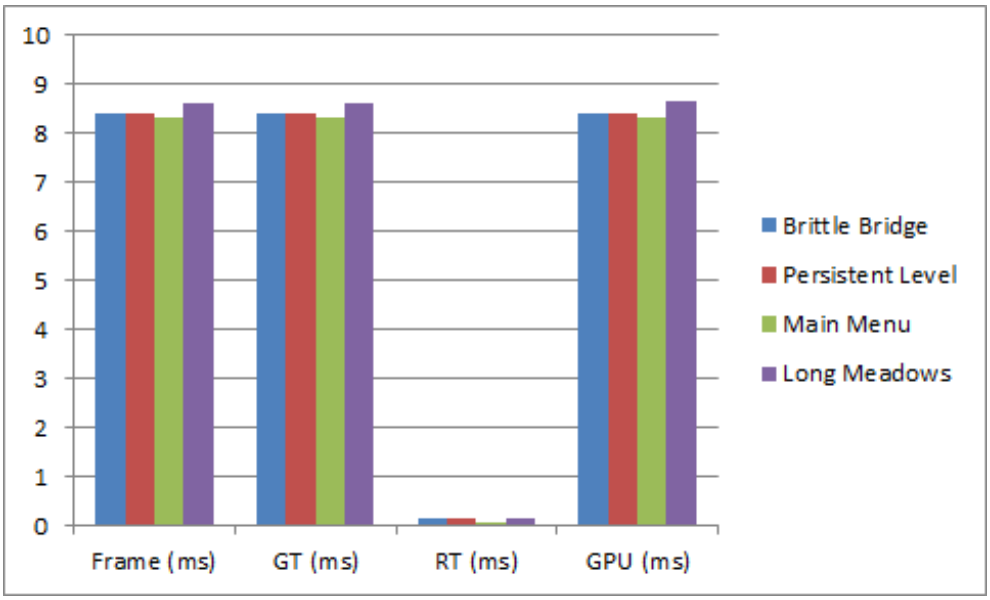
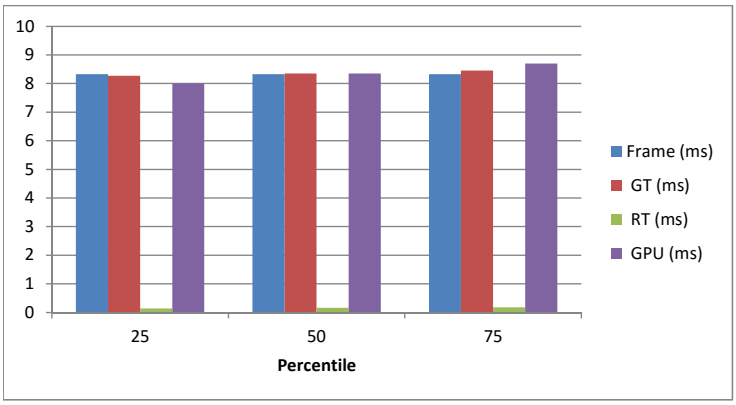
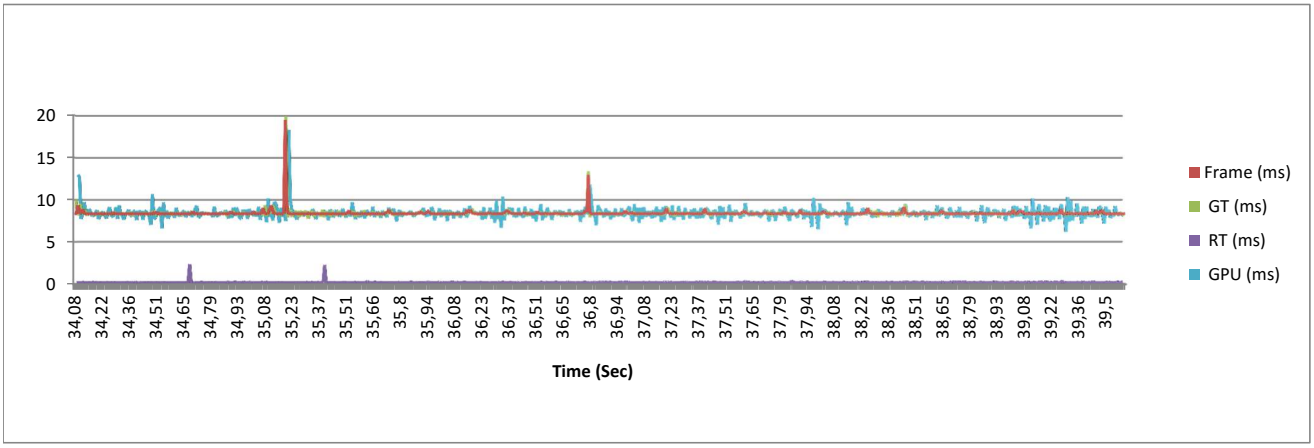


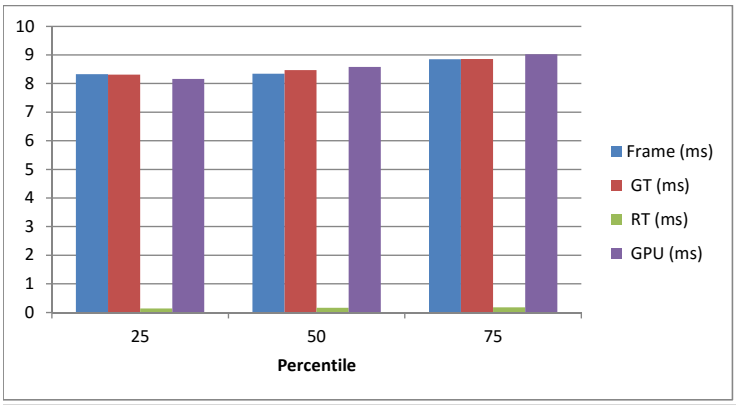
Figure 47: Average time spent loading in each level

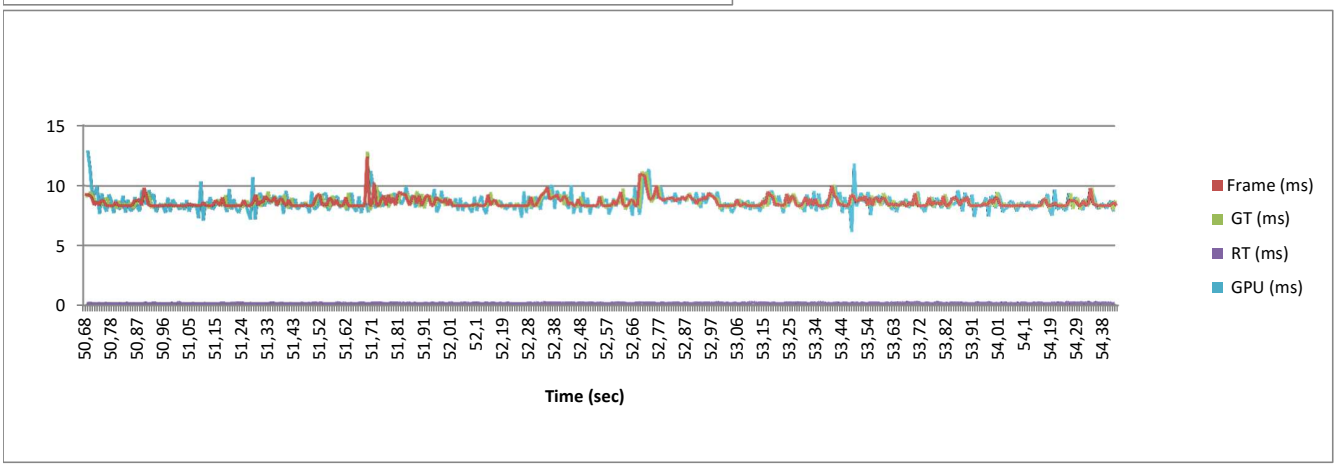
On each page of the four pages below there are two charts. The small chart shows the percentage time spent in milliseconds (ms) on the Game Thread (GT), Render Thread (RT) and Graphics Processing unit (GPU), and the total time spent rendering the image (Frame). The large chart shows the same data but with discrete measurements over a larger period.

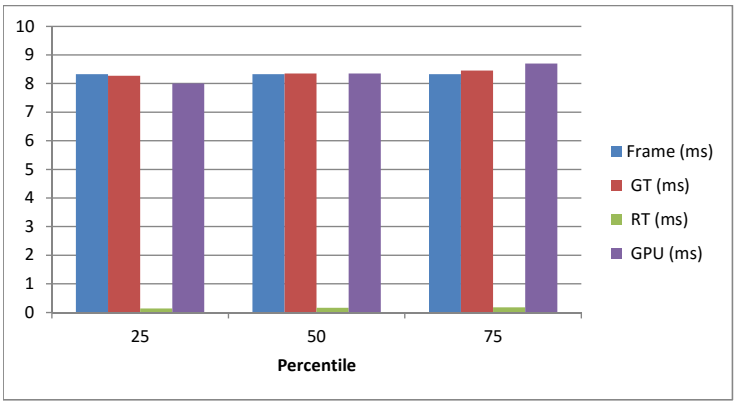
The first page shows the charts for the Brittle Bridge level, the second page for Long Meadows, the third page for Persistent Level (Campaign), and the fourth for Main Menu.

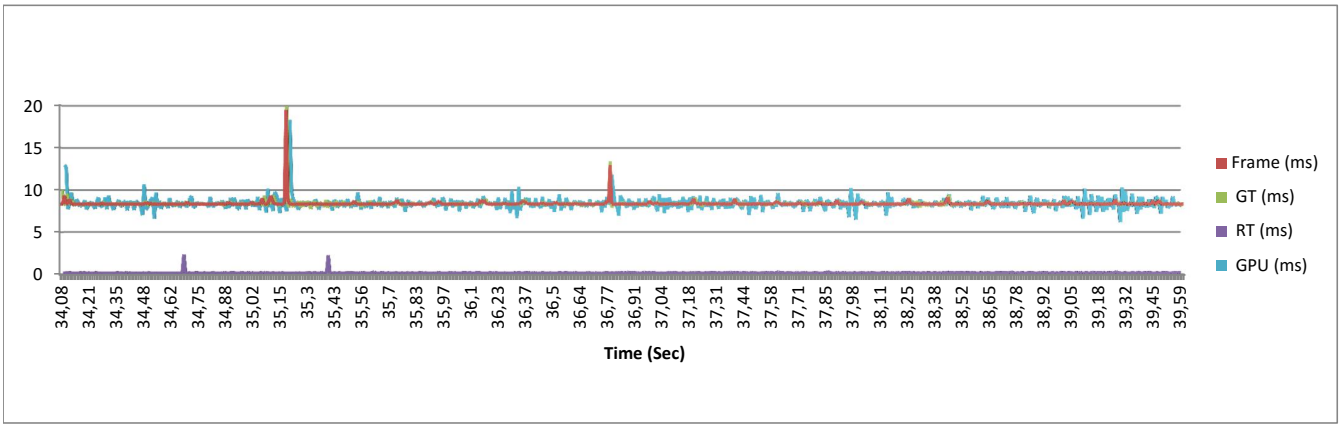

10
9
8
7
6
5
4
3
2
1
0
25
50
75
Percentile
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)

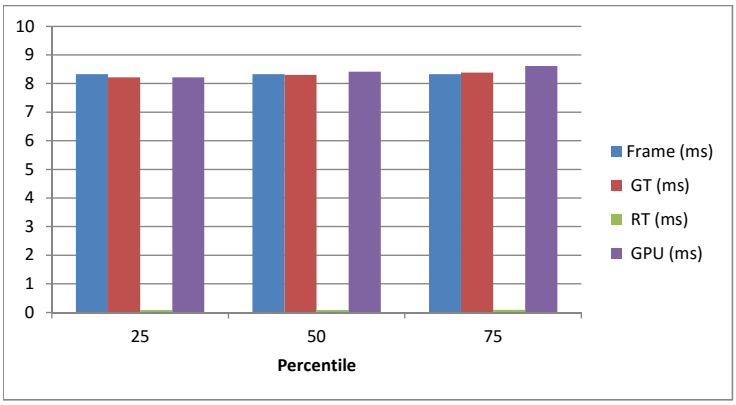

20
15
10
5
0
34,08
34,22
34,36
34,51
34,65
34,79
34,93
35,08
35,23
35,37
35,51
35,66
35,8
35,94
36,08
36,23
36,37
36,51
36,65
36,8
36,94
37,08
37,23
37,37
37,51
37,65
37,79
37,94
38,08
38,22
38,36
38,51
38,65
38,79
38,93
39,08
39,22
39,36
39,5
Time (Sec)
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)

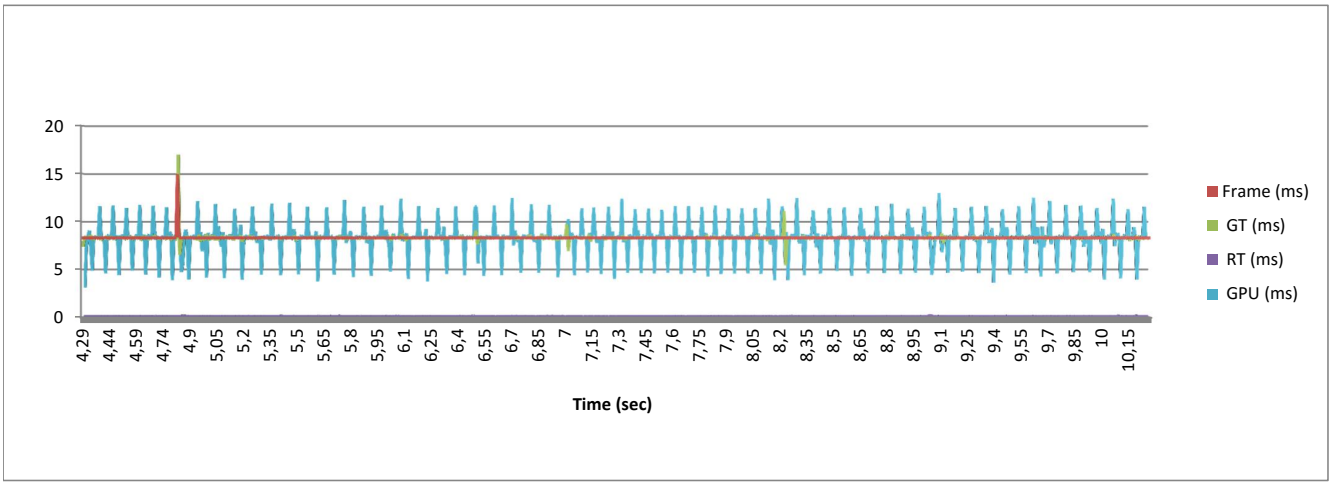

10
9
8
7
6
5
4
3
2
1
0
25
50
75
Percentile
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)


15
10
5
0
50,68
50,78
50,87
50,96
51,05
51,15
51,24
51,33
51,43
51,52
51,62
51,71
51,81
51,91
52,01
52,1
52,19
52,28
52,38
52,48
52,57
52,66
52,77
52,87
52,97
53,06
53,15
53,25
53,34
53,44
53,54
53,63
53,72
53,82
53,91
54,01
54,1
54,19
54,29
54,38
Time (sec)
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)


10
9
8
7
6
5
4
3
2
1
0
25
50
75
Percentile
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)


20
15
10
5
0
34,08
34,21
34,35
34,48
34,62
34,75
34,88
35,02
35,15
35,3
35,43
35,56
35,7
35,83
35,97
36,1
36,23
36,37
36,5
36,64
36,77
36,91
37,04
37,18
37,31
37,44
37,58
37,71
37,85
37,98
38,11
38,25
38,38
38,52
38,65
38,78
38,92
39,05
39,18
39,32
39,45
39,59
Time (Sec)
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)


10
9
8
7
6
5
4
3
2
1
0
25
50
75
Percentile
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)


20
15
10
5
0
4,29
4,44
4,59
4,74
4,9
5,05
5,2
5,35
5,5
5,65
5,8
5,95
6,1
6,25
6,4
6,55
6,7
6,85
7
7,15
7,3
7,45
7,6
7,75
7,9
8,05
8,2
8,35
8,5
8,65
8,8
8,95
9,1
9,25
9,4
9,55
9,7
9,85
10
10,15
Time (sec)
Frame (ms)
GT (ms)
RT (ms)
GPU (ms)

L Survey

The following pages contain the survey document used to ask test-subjects what was missing from the game, what worked, what did not work, etc. converted to PDF format.

noMemTesting Response

Fill in your ratings of our game.

1. **How would you rate the design of the Main menu?**
   Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

2. **How would you rate the design of the Settings menu?**
   Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

3. **How would you rate the design of the Game selection menu?**
   Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

4. **How would you rate the design of the game HUD?**
   Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

5. **How intuitive would you rate the Main menu?**
   Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

6. **How intuitive would you rate the Settings menu?**
   Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

7. **How intuitive would you rate the game selection menu?**
Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

8. **How intuitive would you rate the game hud?**
Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

9. **How fun was the game?**
Markér bare én oval.

	1	2	3	4	5	
Not very fun	◯	◯	◯	◯	◯	very fun

10. **Was the game difficult?**
Markér bare én oval.

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

11. **Did you feel that the game was balanced?**
Markér bare én oval.

	1	2	3	4	5	
Not Very Balanced	◯	◯	◯	◯	◯	Very Balanced

12. **How were the controls?**
Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

13. **How responsive was the game?**
Markér bare én oval.

	1	2	3	4	5	
Not Very	◯	◯	◯	◯	◯	Very

14. **Where the game mechanics easy to understand?**
Markér bare én oval.

	1	2	3	4	5	
Not very	◯	◯	◯	◯	◯	Very

15. **Were you given enough information about each unit?**
Markér bare én oval.

	1	2	3	4	5	
Not enough information	◯	◯	◯	◯	◯	Enough information

16. **How did the game look?**
Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

17. **How would you rate the aesthetics of the game?**
Markér bare én oval.

	1	2	3	4	5	
Very Bad	◯	◯	◯	◯	◯	Very Good

18. **Have you played similar games before**
Markér bare én oval.

- ◯ Yes
- ◯ No
- ◯ Not sure

19. **Would you be interested in playing the game when completed?**
Markér bare én oval.

- ◯ Yes
- ◯ No
- ◯ Not sure

20. **Would you like multiplayer added to the game?**
Markér bare én oval.

- ◯ Yes
- ◯ No
- ◯ Not sure

21. **Any other comments on the game?**

.....

.....

.....

.....

.....

M Survey Results

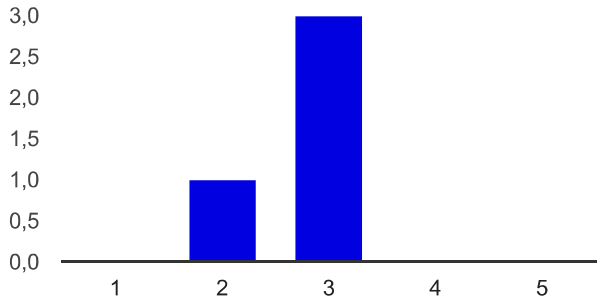
The following pages contain the results of the survey shown in Appendix L.

4 svar

Publiser analytics

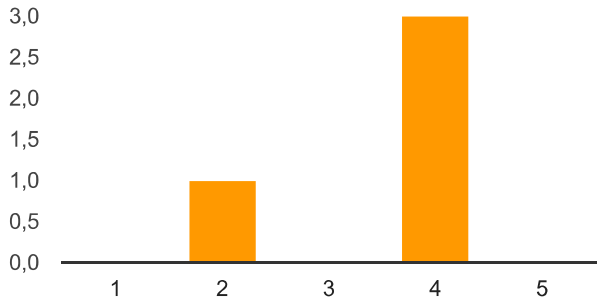
Sammendrag

How would you rate the design of the Main menu?



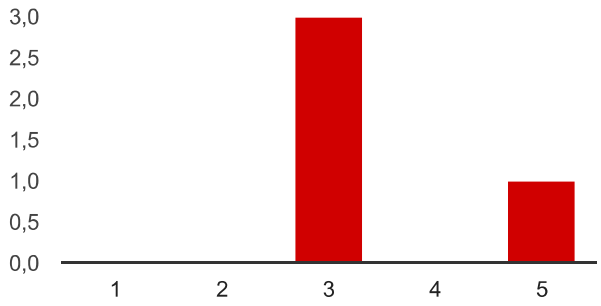
Very Bad:	1	**0**	0 %
	2	**1**	25 %
	3	**3**	75 %
	4	**0**	0 %
Very Good:	5	**0**	0 %

How would you rate the design of the Settings menu?



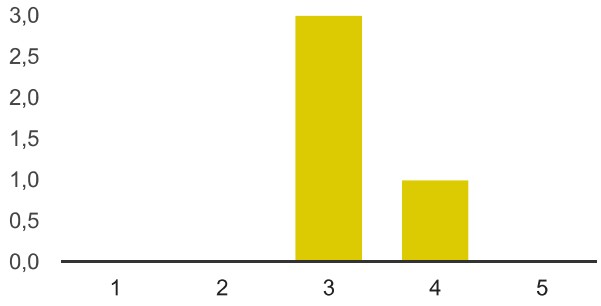
Very Bad:	1	**0**	0 %
	2	**1**	25 %
	3	**0**	0 %
	4	**3**	75 %
Very Good:	5	**0**	0 %

How would you rate the design of the Game selection menu?



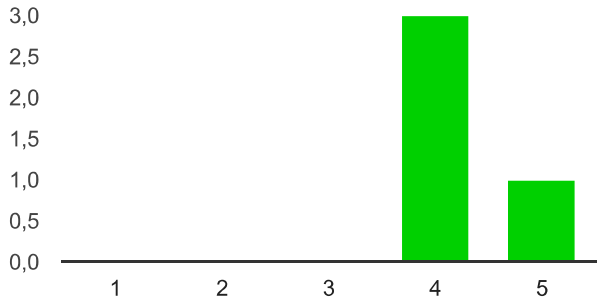
Very Bad:	1	**0**	0 %
	2	**0**	0 %
	3	**3**	75 %
	4	**0**	0 %
Very Good:	5	**1**	25 %

How would you rate the design of the game HUD?

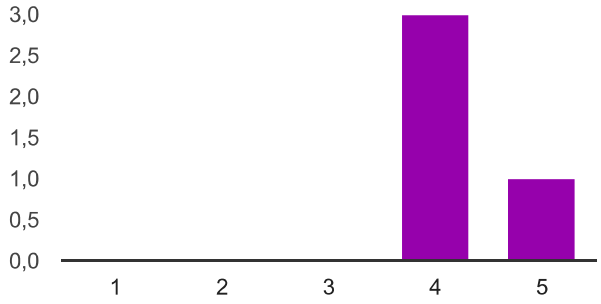


Very Bad:	1	**0**	0 %
	2	**0**	0 %
	3	**3**	75 %
	4	**1**	25 %
Very Good:	5	**0**	0 %

How intuitive would you rate the Main menu?

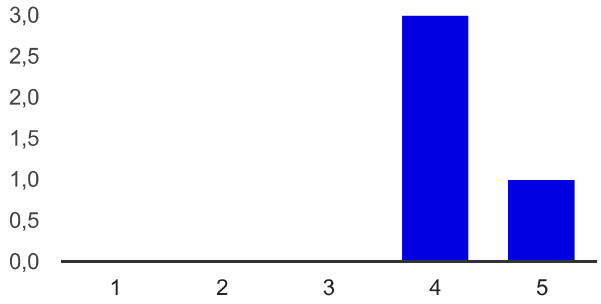


How intuitive would you rate the Settings menu?



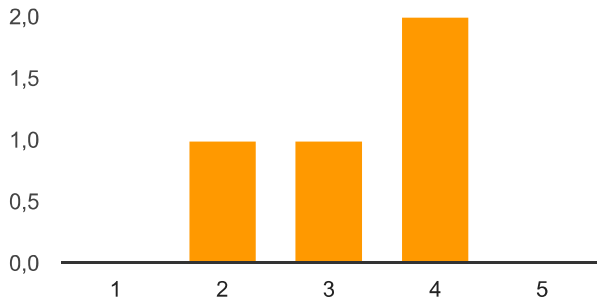
Very Bad:	1	**0**	0 %
	2	**0**	0 %
	3	**0**	0 %
	4	**3**	75 %
Very Good:	5	**1**	25 %

How intuitive would you rate the game selection menu?



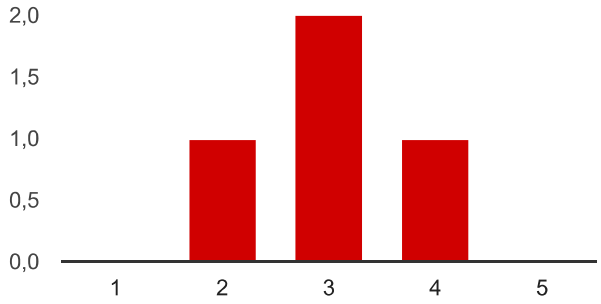
Very Bad:	1	**0**	0 %
	2	**0**	0 %
	3	**0**	0 %
	4	**3**	75 %
Very Good:	5	**1**	25 %

How intuitive would you rate the game hud?



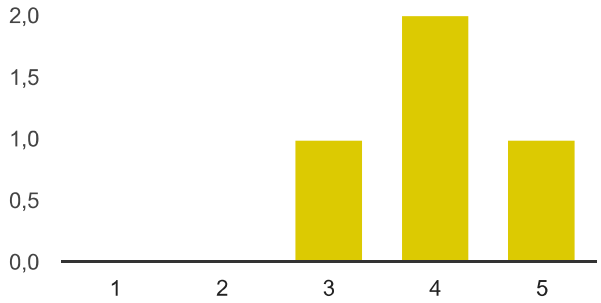
Very Bad:	1	**0**	0 %
	2	**1**	25 %
	3	**1**	25 %
	4	**2**	50 %
Very Good:	5	**0**	0 %

How fun was the game?

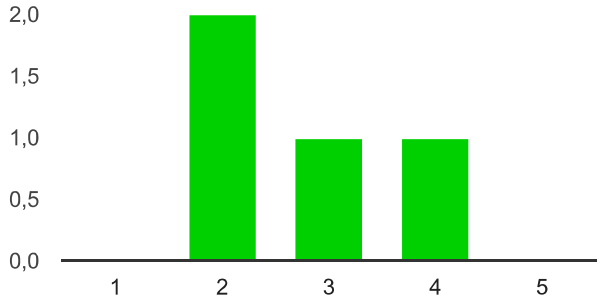


Not very fun:	1	**0**	0 %
	2	**1**	25 %
	3	**2**	50 %
	4	**1**	25 %
very fun:	5	**0**	0 %

Was the game difficult?

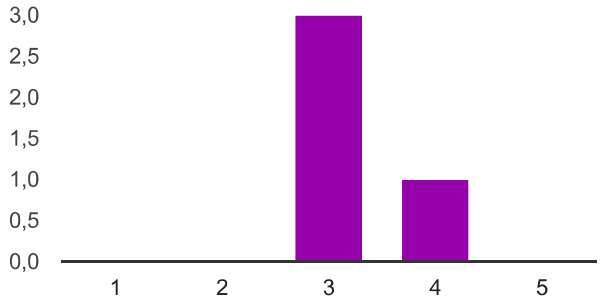


Did you feel that the game was balanced?



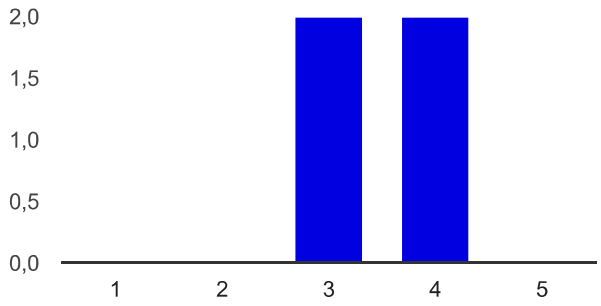
Not Very Balanced:	1	**0**	0 %
	2	**2**	50 %
	3	**1**	25 %
	4	**1**	25 %
Very Balanced:	5	**0**	0 %

How were the controls?



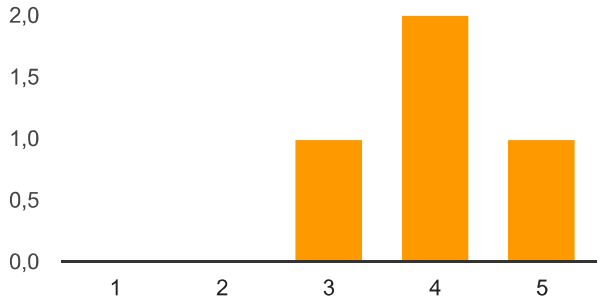
Very Bad:	1	**0**	0 %
	2	**0**	0 %
	3	**3**	75 %
	4	**1**	25 %
Very Good:	5	**0**	0 %

How responsive was the game?



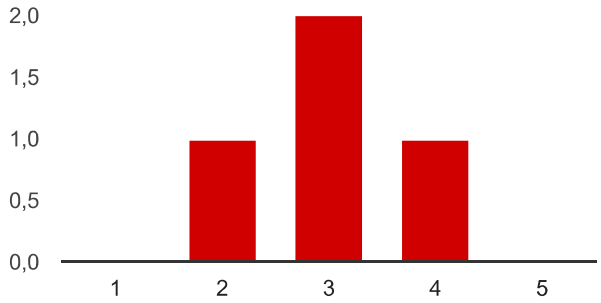
Not Very:	1	**0**	0 %
	2	**0**	0 %
	3	**2**	50 %
	4	**2**	50 %
Very:	5	**0**	0 %

Where the game mechanics easy to understand?

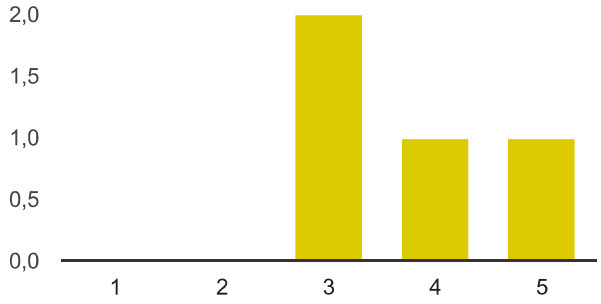


Not very:	1	**0**	0 %
	2	**0**	0 %
	3	**1**	25 %
	4	**2**	50 %
Very:	5	**1**	25 %

Were you given enough information about each unit?

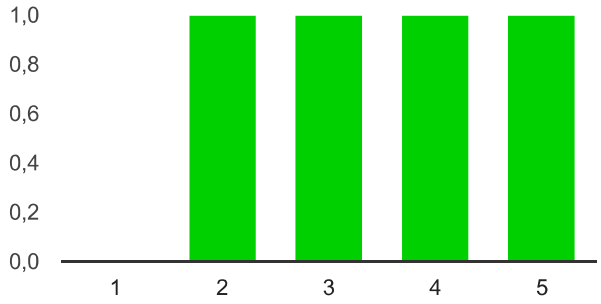


How did the game look?



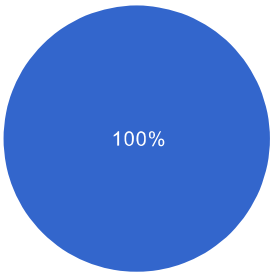
Very Bad:	1	**0**	0 %
	2	**0**	0 %
	3	**2**	50 %
	4	**1**	25 %
Very Good:	5	**1**	25 %

How would you rate the aesthetics of the game?



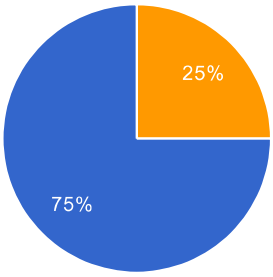
Very Bad:	1	**0**	0 %
	2	**1**	25 %
	3	**1**	25 %
	4	**1**	25 %
Very Good:	5	**1**	25 %

Have you played similar games before



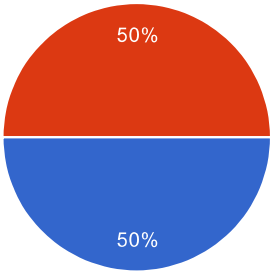
Yes	**4**	100 %
No	**0**	0 %
Not sure	**0**	0 %

Would you be interested in playing the game when completed?



Yes	**3**	75 %
No	**0**	0 %
Not sure	**1**	25 %

Would you like multiplayer added to the game?



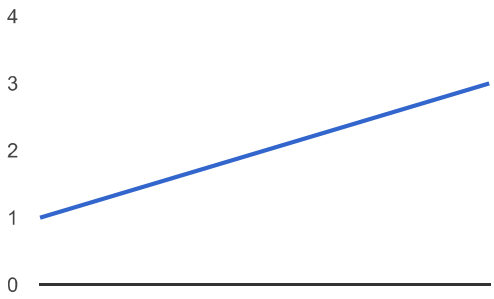
Yes	**2**	50 %
No	**2**	50 %
Not sure	**0**	0 %

Any other comments on the game?

I feel like the camera should be faster and smoother

Immortal grunts YAY!

Antall svar per dag



N UML Statistics

The following is the full version of the UML Statistics:

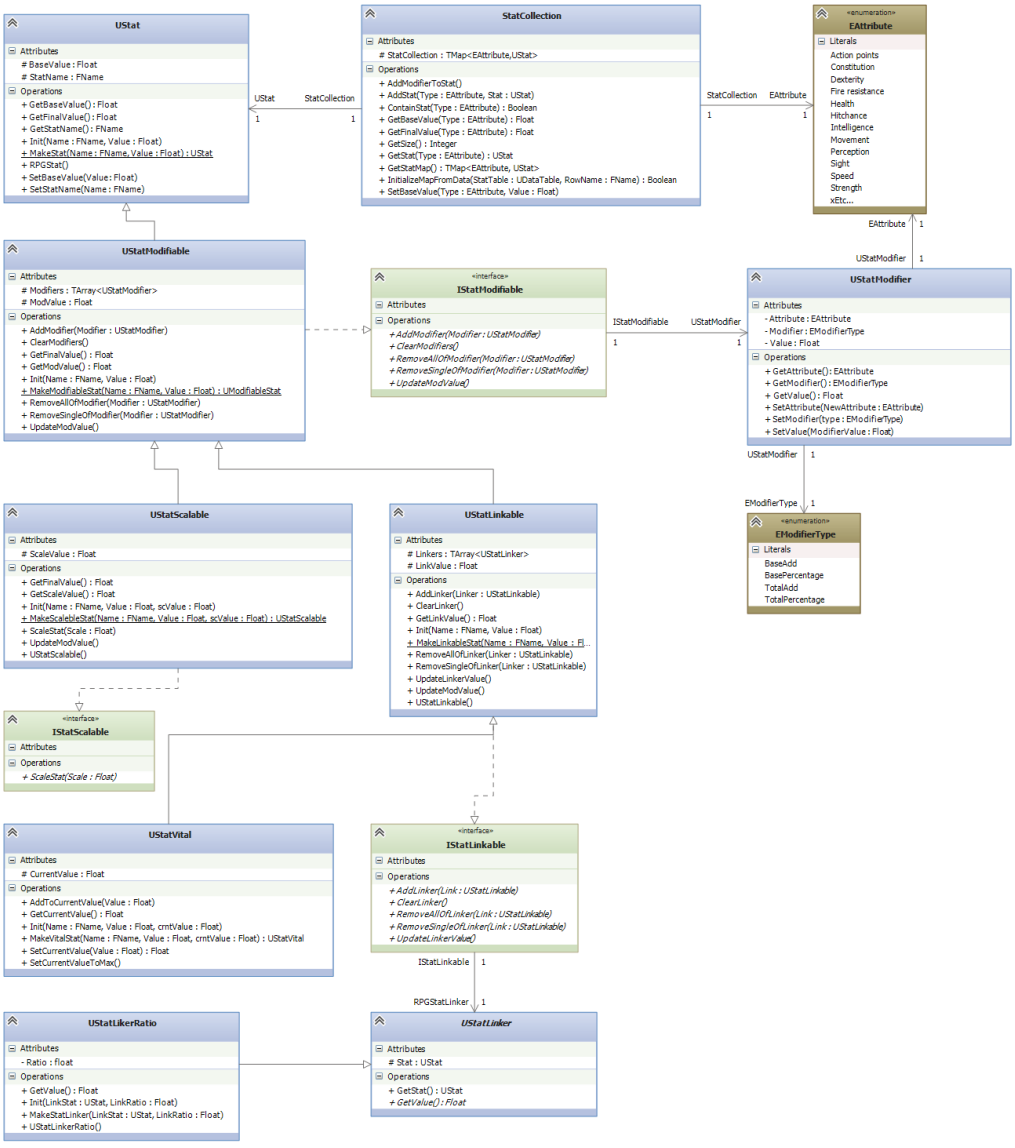


Figure 48: Statistics structure

O Weapons Statistics

The following pages contain the statistics for the weapons used in the game, converted to PDF format.

Weapon stats - Initial values

	BoneAxe	ExecutionerAxe	GoldenAxe	Pulverizer	Temperance	HellHammer	Punctuator	SpikedClub	BlackKnightBlade	Cleaver	Dagger	Fencer	Machete	Sword	FireWand	GruntMasterBlade	SpiderClaw	TrollHammer	TrollRock	GruntPick
WeaponType	Axe	Axe	Axe	Hammer	Hammer	Mace	Mace	Mace	Sword	Sword	Sword	Sword	Sword	Sword	Wand	Sword	Axe	Hammer	Mace	Mace
AttackType	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Ranged	Melee	Melee	Melee	Ranged	Melee
Attribute	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Dexterity	Dexterity	Strength	Strength	Intelligence	Strength	Dexterity	Strength	Strength	Strength
DamageType	Slash	Slash	Slash	Blunt	Blunt	Fire	Blunt	Blunt	Air	Slash	Slash	Slash	Slash	Slash	Fire	Slash	Slash	Earth	Earth	Slash
TwoHanded	No	No	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Range (m)	3	3	3	3	3	3	3	3	3	3	3	3	3	3	10	3	3	4	20	2
MinDamage	10	10	20	25	30	35	30	25	35	10	10	15	20	15	20	25	10	30	20	10
MaxDamage	15	40	25	45	35	55	40	30	55	15	20	25	25	25	40	30	20	40	40	15
APCost	2	4	4	4	4	4	4	4	3	3	2	3	3	3	4	3	2	4	3	2
StatModifer 1	CritChance +10%	CritMulti +50%	Health +10%		Constitution +1	Strength +2	AirResist +30%		AP + 4		CritChance +10%	DodgeChance +15%	Strength +1							
StatModifer 2			Strength +1			FireResist +20%			BlockChance +10%		CritMulti +100%									
StatModifer 3						WaterResist -20%														
StatModifer 4																				
Projectile															Fireball				Rock	

Weapon stats - First Iteration

	BoneAxe	ExecutionerAxe	GoldenAxe	Pulverizer	Temperance	HellHammer	Punctuator	SpikedClub	BlackKnightBlade	Cleaver	Dagger	Fencer	Machete	Sword	FireWand	GruntMasterBlade	SpiderClaw	TrollHammer	TrollRock	GruntPick
WeaponType	Axe	Axe	Axe	Hammer	Hammer	Mace	Mace	Mace	Sword	Sword	Sword	Sword	Sword	Sword	Wand	Sword	Axe	Hammer	Mace	Mace
AttackType	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Melee	Ranged	Melee	Melee	Melee	Ranged	Melee
Attribute	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Strength	Dexterity	Dexterity	Strength	Strength	Intelligence	Strength	Dexterity	Strength	Strength	Strength
DamageType	Slash	Slash	Slash	Blunt	Air	Fire	Blunt	Blunt	Air	Slash	Slash	Slash	Slash	Slash	Fire	Slash	Slash	Earth	Earth	Slash
TwoHanded	No	No	No	Yes	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No
Range (m)	2	2	2	2	2,5	2	2	2	2	2	2	2	2	2	10	2,5	3	3	20	2
MinDamage	10	10	20	25	30	30	40	20	25	10	10	15	10	16	20	20	10	30	20	10
MaxDamage	15	40	30	40	60	40	50	25	35	22	15	20	14	18	40	30	15	40	30	15
APCost	2	4	4	5	6	5	4	4	4	3	2	3	2	3	4	4	2	4	5	2
StatModifer 1	CritChance +10%	Armour +25	Health +10%		Constitution +1	Strength +2	AirResist +30%		AP + 4		CritChance +15%	DodgeChance +15	Strength +1							
StatModifer 2			Strength +1			FireResist +20%			BlockChance +10%		CritMulti +100%									
StatModifer 3						WaterResist -20%														
StatModifer 4																				
Projectile															Fireball				Rock	

P Character Statistics

The following pages contain the statistics for the character in the game in PDF format.

CharacterStats - StartStats

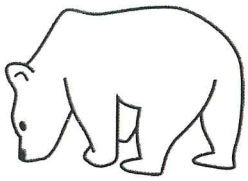
---	Strength	Dexterity	Intelligence	Constitution	Speed	Percepion	Health	ActionPoints	HitChance	CriticalHitChance	CriticalHitMultiplier	Armour	DodgeChance	FireResistence	EarthResistance	WaterResistance	AirResistance	Movement	Sight
Swordsman	6	4	3	10	5	6	50	5	0.4	0.1	2	50	0.05	0	0	0	0	200	3000
Spearman	5	4	3	10	5	7	50	5	0.4	0.1	2	50	0.05	0	0	0	0	200	3000
Barbarian	10	2	2	6	9	5	50	5	0.4	0.1	2	25	0.05	0	0	0	0	200	3000
Gladiator	6	4	3	8	7	6	50	5	0.4	0.1	2	35	0.05	0	0	0	0	200	3000
Knight	8	4	3	12	5	6	50	5	0.4	0.1	2	100	0.05	0.05	0.05	0.05	0.05	200	3000
Paladin	7	4	4	10	6	7	50	5	0.4	0.1	2	100	0.05	0	0.1	0.1	0	200	3000
Champion	10	5	3	14	6	8	50	5	0.4	0.1	2	150	0.05	0.2	0.2	0.2	0.2	200	3000
Scout	2	7	3	6	8	8	50	5	0.4	0.2	3	25	0.1	0	0	0	0	300	3000
Rogue	2	8	3	5	10	10	50	5	0.45	0.2	3	25	0.1	0	0	0	0	300	3000
Assassin	3	10	4	7	10	12	50	5	0.4	0.3	0.35	25	0.1	0	0	0	0	300	3000
Mage	2	2	6	7	5	8	50	5	0.4	0.1	2	25	0.02	0.2	0.2	0.2	0.2	200	3000
Grunt	4	2	2	5	5	5	25	5	0.4	0.05	1.5	35	0.05	-0.2	0	0	0	200	3000
Great_Spider	3	9	1	7	10	6	75	5	0.4	0.2	3	50	0.1	-0.2	0.2	0	0	400	1500
Dark_Knight	10	8	4	14	7	9	100	5	0.4	0.1	2	200	0.05	0.35	0.5	-0.3	-0.3	200	3000
Troll	12	2	1	14	4	2	75	5	0.4	0.1	2	75	0	0.5	0.5	-0.2	-0.2	200	3000
Half_Giant	6	4	3	12	5	8	50	5	0.4	0.1	2	50	0.05	-0.2	0	0	0	200	3000
Grunt_Master	7	3	3	10	6	6	50	5	0.4	0.1	2	60	0.05	0	0	0	0	300	3000

CharacterStats - FirstBalanceStats

---	Strength	Dexterity	Intelligence	Constitution	Speed	Percepion	Health	ActionPoints	HitChance	CriticalHitChance	CriticalHitMultiplier	Armour	DodgeChance	FireResistence	EarthResistance	WaterResistance	AirResistance	Movement	Sight
Swordsman	6	4	3	10	7	6	50	5	0.5	0.1	2	50	0.05	0	0	0	0	200	3000
Barbarian	10	2	2	6	9	5	50	5	0.5	0.1	2	0	0.05	0	0	0	0	200	3000
Gladiator	6	4	3	8	7	6	50	5	0.5	0.1	2	35	0.05	0	0	0	0	200	3000
Knight	8	4	3	12	7	6	50	5	0.5	0.1	2	100	0.05	0.05	0.05	0.05	0.05	200	3000
Paladin	7	4	4	10	7	7	50	5	0.5	0.1	2	100	0.05	0	0.1	0.1	0	200	3000
Champion	10	5	3	14	8	8	50	5	0.5	0.1	2	150	0.05	0.2	0.2	0.2	0.2	200	3000
Scout	2	7	3	6	10	8	50	5	0.6	0.2	3	25	0.1	0	0	0	0	300	3000
Rogue	2	8	3	5	10	10	50	5	0.6	0.2	3	25	0.1	0	0	0	0	300	3000
Assassin	3	10	4	7	10	12	50	5	0.6	0.3	3.5	25	0.1	0	0	0	0	300	3000
Mage	2	2	6	7	8	8	50	5	0.5	0.1	2	25	0.02	0.2	0.2	0.2	0.2	200	3000
Grunt	4	2	2	5	6	5	25	5	0.5	0.05	1.5	35	0.05	-0.2	0	0	0	200	3000
Half_Giant	6	4	3	12	6	8	50	5	0.5	0.1	2	50	0.05	-0.2	0	0	0	200	3000
Grunt_Master	7	3	3	10	6	6	50	5	0.5	0.1	2	60	0.05	0	0	0	0	300	3000
Great_Spider	3	9	1	7	10	6	50	5	0.6	0.2	3	50	0.1	-0.2	0.2	0	0	400	1500
Troll	12	2	1	14	6	2	100	5	0.5	0.1	2	75	0	0.5	0.5	-0.2	-0.2	200	3000
Dark_Knight	10	8	4	14	7	9	75	5	0.5	0.1	2	200	0.05	0.35	0.5	-0.3	-0.3	200	3000

Q Responsibilities and roles accurately explained

The following two pages contain more accurate explanations for responsibilities and roles using the animal method introduced to us by Simon McCallum, converted to PDF format.

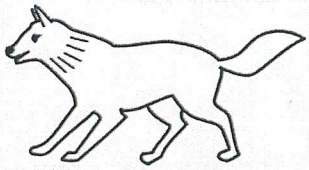


Bear

The Leader

The person who likes to lead, has a vision and ideas they want to develop. Usually the loudest in the group.

Achieve
Control

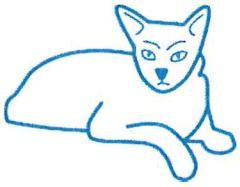


Wolf

The Manager

The head of the pack. Makes sure everyone is contributing, and tells the Bear to stop talking, if needed.

Belong
Control



Cat

The Cynic

The person who can find the holes in ideas, and the ways of cheating the system

Control
Achieve

Puppy

The Enthusiast

Always supports ideas, and wants to help add and develop ideas.

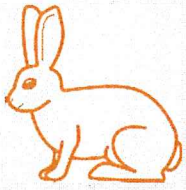
Belong
Achieve

Owl

The Processor

Focuses on making sure that the required objectives are met. Likes to make and complete lists.

Control
Belong



Rabbit

The Facilitator

Spots when things are missing and volunteers to go and get stuff. Talks to the session organiser

Achieve
Belong

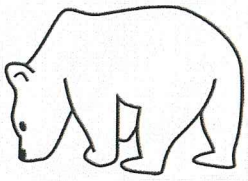
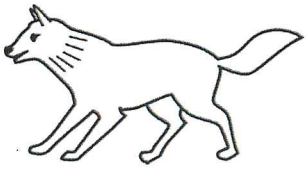
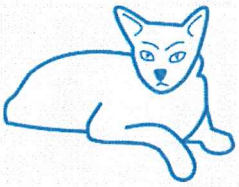
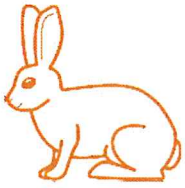
McClelland's Three Needs Theory

Achieve.	Need for Achievement (N-ach)	They want to see something actually get made or succeed.
Belong.	Need for Affiliation (N-aff)	Social acceptance and being part of a group.
Control.	Need for Power (N-pow)	Status, influence and power. Controlling a situation.

Group formation

Every participant chooses an animal avatar. The bears need to feel ownership of the idea that is being developed. The groups are then formed with one avatar type per group. In groups smaller than 6 some people will take on two roles. Larger than 6 will have more Cats, Dogs and Rabbits.

The use of animal avatars creates cognitive distance between the role in the group and the individual. This helps with group organisation and communication.

Animal		Action
	Bear	When no one is speaking, the bear needs to start the conversation. They are to have the overall vision for the project.
	Wolf	If the bear is talking to much the wolf must stop the bear and get other people in the group to talk. Potentially taking note of who is contributing and ensuring that everyone in the group has a chance to talk.
	Cat	Find problems that when fixed will make the project better. Find potential road blocks and flag them early so they can be resolved or used to kill bad ideas. Should not try to kill idea too early.
	Puppy	Always enthusiastic. Balances the Cat by providing potential solutions for challenges identified by the Cat. Listens and supports the creation of new ideas and innovative solutions to challenges.
	Owl	Takes notes and watches the schedule. Allows some variation in timing, but uses phases like: "I hate to have to be the owl but we have to move to the next discussion point"
	Rabbit	Watches for problems in the group where people are missing something. Also thinks about the infrastructure around a project and what other things need to be in place to make an idea work.

Remember: the objective is to help the group progress.

All the roles are important and no one should feel that they are in any way less important to the success of the group than any other person.

R Doxygen Reference Manual

The following pages contain the Doxygen reference manual output for C++ code written in our project, converted to PDF format.

noMemory

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

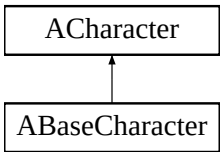
Chapter 3

Class Documentation

3.1 ABaseCharacter Class Reference

`#include <BaseCharacter.h>`

Inheritance diagram for ABaseCharacter:



Public Member Functions

- **ABaseCharacter** (const FObjectInitializer &ObjectInitializer)
- virtual void **BeginPlay** () override
- virtual void **Tick** (float DeltaSeconds) override
- virtual void **SetupPlayerInputComponent** (class UInputComponent *InputComponent) override
- virtual float **TakeDamage** (float DamageAmount, struct FDamageEvent const &DamageEvent, class AController *EventInstigator, class AActor *DamageCauser) override
- virtual bool Initialize (UDataTable *StatTable)
- void AttackUnit (ABaseCharacter *Attacking)
- bool HitTarget (ABaseCharacter *Target)
- void CharacterDeath (float Delay)
- float CalculateReducedDamage (float RawDamage, struct FDamageEvent const &DamageEvent)
- void EquipWeapon (TSubclassOf< ABaseWeapon > Weapon)
- void UnEquipWeapon (ABaseWeapon *Weapon)
- void GainXP (int32 XP)
- void Select ()
- void DeSelect ()
- void BeginTurn ()
- void MoveDistance (float Distance)
- bool IsMoving ()
- bool IsInAnimation ()
- bool IsPossibleToHit (ABaseCharacter *TargetUnit)
- void DisplayFloatingText (const FString &Text)

- void AddToCombatLog (const FString &Line)
- void RotateToFaceLocation (FVector Location)
- void OutlineEnemiesInRange ()
- int32 GetTeam ()
- FName GetUnitName ()
- UStatCollection * GetStatCollection ()
- float GetMinDamage ()
- float GetMaxDamage ()
- float GetActionPoints ()
- float GetAttackCost ()
- float GetMoveCost (float Distance)
- float CalculateHitChance ()
- float GetChanceToHitCharacter (ABaseCharacter *Character)
- float GetRange ()
- TSubclassOf< UBaseDamageType > GetDamageType ()
- void SetStatCollection (UStatCollection *StatCollection)
- void SetTeam (int32 NewTeam)
- UDecalComponent * GetTeamDecal ()
- void SetTeamDecalColor (FLinearColor Color)
- UDecalComponent * GetRangeDecal ()
- void SetRangeDecalColor (FLinearColor Color)

Protected Attributes

- FName **UnitName**
- int32 **Team**
- TSubclassOf< UStatCollection > **CollectionClass**
- UStatCollection * **Collection**
- float **UnUsedMovement**
- bool **bAttacking**
- bool **bHit**
- bool **bDead**
- FName **MainHandSocket**
- FName **OffHandSocket**
- TSubclassOf< ABaseWeapon > **MainHandWeaponClass**
- ABaseWeapon * **MainWeapon**
- TSubclassOf< ABaseWeapon > **OffhandWeaponClass**
- ABaseWeapon * **OffHandWeapon**
- UDecalComponent * **TeamDecal**
- UDecalComponent * **RangeDecal**

3.1.1 Detailed Description

BaseCharacter for the units in the game.

See also

ACharacter

3.1.2 Member Function Documentation

3.1.2.1 void ABaseCharacter::AddToCombatLog (const FString & *Line*)

Sends text line to combat log

Parameters

Line	to add to combat log

3.1.2.2 void ABaseCharacter::AttackUnit (ABaseCharacter * *Attacking*)

Attack another character

Parameters

Attacking	this character

3.1.2.3 void ABaseCharacter::BeginTurn ( )

Called at the start of each teams turn

3.1.2.4 float ABaseCharacter::CalculateHitChance ( )

Get the hitchance after applying weapon attribute

Returns

hit chance

3.1.2.5 float ABaseCharacter::CalculateReducedDamage (float *RawDamage*, struct FDamageEvent const & *DamageEvent*)

Calculate damage reduction from stats such as armour and resistances

Parameters

RawDamage	before reduction
DamageEvent	used to calculate the reduced damage

Returns

Reduced damage

3.1.2.6 void ABaseCharacter::CharacterDeath (float *Delay*)

Called to remove character makes sure that other attached actors also are removed

Parameters

Delay	to wait before removing

3.1.2.7 void ABaseCharacter::DeSelect ( )

called when being DeSelected

3.1.2.8 void ABaseCharacter::DisplayFloatingText (const FString & *Text*)

Blueprint event used to display information over actor

Parameters

Distance	to move

3.1.2.9 void ABaseCharacter::EquipWeapon (TSubclassOf< ABaseWeapon > *Weapon*)

Equipe a Weapon to the character

Parameters

Weapon	to equip

3.1.2.10 void ABaseCharacter::GainXP (int32 *XP*)

Character gains experience

Parameters

XP	gained

3.1.2.11 float ABaseCharacter::GetActionPoints ( )

Get the amount of action points available to the character

Returns

action points

3.1.2.12 float ABaseCharacter::GetAttackCost ( )

Get the amount of action points needed to attack with weapons

Returns

action points

3.1.2.13 float ABaseCharacter::GetChanceToHitCharacter (ABaseCharacter * *Character*)

Get the chance to hit ABaseCharacter

Returns

chance to hit

3.1.2.14 TSubclassOf< UBaseDamageType > ABaseCharacter::GetDamageType ( )

Get the damage type for the currently equiped weapons

Returns

Damage type of the main weapon else damage type of the offhand weapon

3.1.2.15 float ABaseCharacter::GetMaxDamage ( )

Get maximum damage done by weapons

Returns

max damage

3.1.2.16 float ABaseCharacter::GetMinDamage ( )

Get minimum damage done by weapons

Returns

minimum damage

3.1.2.17 float ABaseCharacter::GetMoveCost (float *Distance*)

Get the amount of action points needed to move that distance

Parameters

Distance	to move

Returns

action points

3.1.2.18 float ABaseCharacter::GetRange ( )

Get the range for the longest weapon

Returns

Range

3.1.2.19 UDecalComponent * ABaseCharacter::GetRangeDecal ( )

Get reference to team decal

Returns

Decal reference

3.1.2.20 UStatCollection * ABaseCharacter::GetStatCollection ( )

Get stat collection

Returns

statCollection or null if there is none

3.1.2.21 int32 ABaseCharacter::GetTeam ( )

Get units team number

Returns

team number

3.1.2.22 UDecalComponent * ABaseCharacter::GetTeamDecal ( )

Get reference to team decal

Returns

Decal reference

3.1.2.23 FName ABaseCharacter::GetUnitName ( )

Get units name

Returns

unit name

3.1.2.24 bool ABaseCharacter::HitTarget (ABaseCharacter * *Target*)

Check if we can hit our target

Parameters

Attacking	this character

Returns

true if hit target

3.1.2.25 bool ABaseCharacter::Initialize (UDataTable * *StatTable*) [virtual]

Initializes the values of the character using a stat table

Parameters

StatTable	containing all the different stats needed to initialize the character

Returns

true if initialized, false if not

3.1.2.26 bool ABaseCharacter::IsInAnimation ( )

Check if the unit is performing an animation

Returns

true if in any other animation than idle

3.1.2.27 bool ABaseCharacter::IsMoving ( )

Check if the unit is moving

Returns

true if moving

3.1.2.28 bool ABaseCharacter::IsPossibleToHit (ABaseCharacter * *TargetUnit*)

Check if it is possible to hit unit from the possition we are standing

Parameters

TargetUnit	to check if possible to hit

Returns

true if it is possible

3.1.2.29 void ABaseCharacter::MoveDistance (float *Distance*)

Needs to be called when the unit moves.

Parameters

Distance	to move

3.1.2.30 void ABaseCharacter::OutlineEnemiesInRange ( )

Findes enemies that are in attack range and enable there outline

3.1.2.31 void ABaseCharacter::RotateToFaceLocation (FVector *Location*)

Make the unit face a chosen actor

Parameters

Actor	to face

3.1.2.32 void ABaseCharacter::Select ( )

called when being selected

3.1.2.33 void ABaseCharacter::SetRangeDecalColor (FLinearColor *Color*)

Set the color of the Range decal

Parameters

Color	to set to

3.1.2.34 void ABaseCharacter::SetStatCollection (UStatCollection * *StatCollection*)

Set the current stat collection to a new one

Parameters

StatCollection	to set to

3.1.2.35 void ABaseCharacter::SetTeam (int32 *NewTeam*)

Change the units team

Returns

NewTeam to join

3.1.2.36 void ABaseCharacter::SetTeamDecalColor (FLinearColor *Color*)

Set the color of the Team Decal

Parameters

Color	to set to

3.1.2.37 void ABaseCharacter::UnEquipWeapon (ABaseWeapon * *Weapon*)

Unquips an Weapon from the character

Parameters

Weapon	to add to chatacter

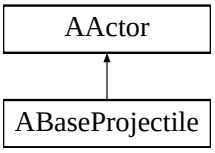
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Character/BaseCharacter.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/Character/BaseCharacter.cpp

3.2 ABaseProjectile Class Reference

```
#include <BaseProjectile.h>
```

Inheritance diagram for ABaseProjectile:



Public Member Functions

- virtual void **BeginPlay** () override
- virtual void **Tick** (float DeltaSeconds) override
- float GetDamage ()
- void SetDamage (float NewDamage)

Protected Attributes

- UProjectileMovementComponent * **MovementComponent**
- USphereComponent * **CollisionComponent**
- float **Damage**
- TSubclassOf< UBaseDamageType > **DamageType**

3.2.1 Detailed Description

Base Actor for Projectiles in the game

See also

AActor

3.2.2 Member Function Documentation

3.2.2.1 float ABaseProjectile::GetDamage ( )

Get the damage of the projectile

Returns

damage

3.2.2.2 void ABaseProjectile::SetDamage (float *NewDamage*)

Set the damage of the projectile to the new value

Parameters

NewDamage	to set damage to

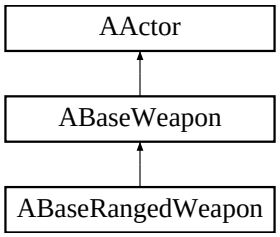
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Actor/BaseProjectile.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/Actor/BaseProjectile.cpp

3.3 ABaseRangedWeapon Class Reference

`#include <BaseRangedWeapon.h>`

Inheritance diagram for ABaseRangedWeapon:



Public Member Functions

- TSubclassOf< ABaseProjectile > GetProjectileClass ()

Protected Attributes

- TSubclassOf< ABaseProjectile > **ProjectileClass**

3.3.1 Detailed Description

Base Actor for ranged weapons in the game.

See also

ABaseWeapon

3.3.2 Member Function Documentation

3.3.2.1 TSubclassOf< ABaseProjectile > ABaseRangedWeapon::GetProjectileClass ( )

Get the projectile class created by the weapon

Returns

sub class of ABaseProjectile or null if none have been set

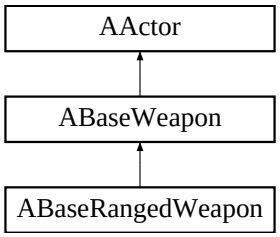
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Actor/BaseRangedWeapon.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/Actor/BaseRangedWeapon.cpp

3.4 ABaseWeapon Class Reference

`#include <BaseWeapon.h>`

Inheritance diagram for ABaseWeapon:



Public Member Functions

- **ABaseWeapon** (const FObjectInitializer &ObjectInitializer)
- virtual void **BeginPlay** () override
- virtual void **Tick** (float DeltaSeconds) override
- EAttackType GetAttackType ()
- TArray< UStatModifier * > GetStatModifiers ()
- TSubclassOf< UBaseDamageType > GetDamageType ()
- EAttribute GetAttributeType ()
- bool GetTwohanded ()
- float GetRange ()
- float GetMinDamage ()
- float GetMaxDamage ()
- float GetAPCost ()

Protected Attributes

- FName **WeaponName**
- EWeaponType **WeaponType**
- EAttackType **AttackType**
- EAttribute **AttributeType**
- TSubclassOf< UBaseDamageType > **DamageType**
- TArray< UStatModifier * > **StatModifiers**
- bool **bTwoHanded**
- float **Range**
- float **MinDamage**
- float **MaxDamage**
- float **APCost**

3.4.1 Detailed Description

Base actor for the weapons in the game.

See also

AActor

3.4.2 Member Function Documentation

3.4.2.1 float ABaseWeapon::GetAPCost ( )

Get the amount of action points needed for weapons to attack

Returns

Action points

3.4.2.2 EAttackType ABaseWeapon::GetAttackType ( )

Get the attack type of the weapon

Returns

AttackType

3.4.2.3 EAttribute ABaseWeapon::GetAttributeType ( )

Get the attribute type the weapon uses

Returns

EAttribute

3.4.2.4 TSubclassOf< UBaseDamageType > ABaseWeapon::GetDamageType ( )

Get the type of damage the weapon does

Returns

sub class of UBaseDamageType

3.4.2.5 float ABaseWeapon::GetMaxDamage ( )

Get the maximum amount of damage the weapon does

Returns

maximum damage

3.4.2.6 float ABaseWeapon::GetMinDamage ( )

Get the minimum amount of damage the weapon does

Returns

minimum damage

3.4.2.7 float ABaseWeapon::GetRange ( )

Get The range of the weapon

Returns

weapon range (cm)

3.4.2.8 TArray< UStatModifier * > ABaseWeapon::GetStatModifiers ( )

Get all the modifers in the weapon

Returns

Array of modifiers

3.4.2.9 bool ABaseWeapon::GetTwohanded ( )

Get if the weapon is twohanded

Returns

true if twohanded

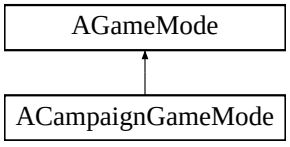
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Actor/BaseWeapon.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/Actor/BaseWeapon.cpp

3.5 ACampaignGameMode Class Reference

`#include <CampaignGameMode.h>`

Inheritance diagram for ACampaignGameMode:



Public Member Functions

- bool LoadCampaignSettings ()

Protected Attributes

- EDifficulty **Difficulty**

3.5.1 Detailed Description

Game mode for campaign matches

See also

AGameMode

3.5.2 Member Function Documentation

3.5.2.1 bool ACampaignGameMode::LoadCampaignSettings ( )

Load game mode setting from file

Returns

True if successfully loaded

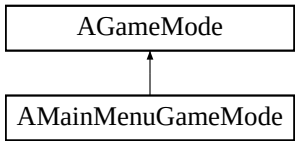
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/GameMode/Campaign↩ GameMode.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/GameMode/Campaign↩ GameMode.cpp

3.6 AMainMenuGameMode Class Reference

`#include <MainMenuGameMode.h>`

Inheritance diagram for AMainMenuGameMode:



Public Member Functions

- void SaveCampaignSettings ()
- void SaveSkirmishSettings ()

Protected Attributes

- EDifficulty **Difficulty**
- FString **SelectedLevel**

3.6.1 Detailed Description

Game mode used in main menu, made to setup the next game mode

3.6.2 Member Function Documentation

3.6.2.1 void AMainMenuGameMode::SaveCampaignSettings ( )

Save campaign settings

3.6.2.2 void AMainMenuGameMode::SaveSkirmishSettings ( )

Save skirmish settings

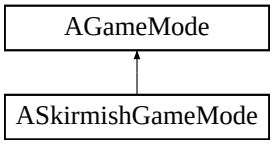
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/GameMode/MainMenu↩ GameMode.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/GameMode/MainMenu↩ GameMode.cpp

3.7 ASkirmishGameMode Class Reference

```
#include <SkirmishGameMode.h>
```

Inheritance diagram for ASkirmishGameMode:



Public Member Functions

- virtual void **Tick** (float DeltaSeconds)
- bool EndGameCondition ()
- void StartGame ()
- void NextTurn ()
- int32 GetNumberOfTeams ()
- EDifficulty GetDifficulty ()
- int32 GetCurrrentTeam ()
- int32 GetPlayerTeam ()
- void SetNumberOfTeams (int32 TeamCount)
- void SetDifficulty (EDifficulty NewDifficulty)
- void SetCurrrentTeam (int32 NewCurrentTeam)
- void SetPlayerTeam (int32 NewPlayersTeam)
- bool LoadSkirmishSettings ()
- void OnPlayerWin ()
- void OnPlayerLose ()

3.7.1 Detailed Description

Game mode for skirmish matches

See also

AGameMode

3.7.2 Member Function Documentation

3.7.2.1 bool ASkirmishGameMode::EndGameCondition ( )

Start next turn

Returns

true if end game condition are met

3.7.2.2 int32 ASkirmishGameMode::GetCurrrentTeam ( )

Get the team that is currently taking there turn

Returns

Team thats taking there turn

3.7.2.3 EDifficulty ASkirmishGameMode::GetDifficulty ( )

Get the difficulty of the game mode

Returns

Difficulty

3.7.2.4 int32 ASkirmishGameMode::GetNumberOfTeams ( )

Get the amount of teams in the game

Returns

Amount of teams

3.7.2.5 int32 ASkirmishGameMode::GetPlayerTeam ( )

Get the team the player is controlling

Returns

Player controlled team number

3.7.2.6 bool ASkirmishGameMode::LoadSkirmishSettings ( )

Load game mode setting from file

Returns

True if successfully loaded

3.7.2.7 void ASkirmishGameMode::NextTurn ( )

Start next turn

3.7.2.8 void ASkirmishGameMode::OnPlayerLose ( )

Blueprint Event, called when player loses

3.7.2.9 void ASkirmishGameMode::OnPlayerWin ( )

Blueprint Event, called when player wins

3.7.2.10 void ASkirmishGameMode::SetCurrrentTeam (int32 *NewCurrentTeam*)

Set the currently playing team

Parameters

NewCurrentTeam	to take there turn

3.7.2.11 void ASkirmishGameMode::SetDifficulty (EDifficulty *NewDifficulty*)

Set new difficulty

Parameters

NewDifficulty	to set to

3.7.2.12 void ASkirmishGameMode::SetNumberOfTeams (int32 *TeamCount*)

Set the number of teams in the game

Parameters

TeamCount	to set to

3.7.2.13 void ASkirmishGameMode::SetPlayerTeam (int32 *NewPlayersTeam*)

Set the team the player is controlling

Parameters

NewPlayersTeam	to controll

3.7.2.14 void ASkirmishGameMode::StartGame ( )

Start the game by starting a players turn

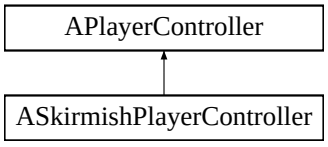
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/GameMode/Skirmish↩ GameMode.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/GameMode/Skirmish↩ GameMode.cpp

3.8 ASkirmishPlayerController Class Reference

`#include <SkirmishPlayerController.h>`

Inheritance diagram for ASkirmishPlayerController:



Public Member Functions

- virtual void **BeginPlay** () override
- virtual void **Tick** (float DeltaSeconds) override
- virtual void **SetupInputComponent** () override
- void BeginTurn ()
- void SelectReleased ()
- void ActionReleased ()
- void ToggleTeamColor ()
- void ToggleRange ()
- void SelectUnit (ABaseCharacter *NewCharacter)
- bool DeSelectUnit ()
- int32 GetTeam ()
- USkirmishWidget * GetSkirmishUI ()
- void SetTeam (int32 NewTeam)
- void OnBeginTurn ()
- void OnSelect ()
- void OnDeSelect ()
- void OnAttack (const ABaseCharacter *Defender)
- void OnMove (const FVector &Location)
- void OnOverUnit (ABaseCharacter *Unit)
- void OnOverLocation (const FVector &Location)

Protected Attributes

- TSubclassOf< USkirmishWidget > **SkirmishWidget**
- USkirmishWidget * **SkirmishUI**
- bool **bTurn**
- int32 **Team**
- ABaseCharacter * **SelectedCharacter**

3.8.1 Detailed Description

Player Controller for Skirmish games

See also

APlayerController

3.8.2 Member Function Documentation

3.8.2.1 void ASkirmishPlayerController::ActionReleased ( )

Released action button, moves or attacks using the selected unit

3.8.2.2 void ASkirmishPlayerController::BeginTurn ( )

Called to start the players turn

3.8.2.3 bool ASkirmishPlayerController::DeSelectUnit ( )

Deselect the current selected character

Returns

false if not possible to deselect

3.8.2.4 USkirmishWidget * ASkirmishPlayerController::GetSkirmishUI ( )

Get the skirmish UI

Returns

the skirmish ui or null if there is none

3.8.2.5 int32 ASkirmishPlayerController::GetTeam ( )

Get controllers team number

Returns

team number

3.8.2.6 void ASkirmishPlayerController::OnAttack (const ABaseCharacter * *Defender*)

Blueprint Event, called when attacking a unit

Parameters

Defender	from the attack

3.8.2.7 void ASkirmishPlayerController::OnBeginTurn ( )

Blueprint Event, called when begining our turn

3.8.2.8 void ASkirmishPlayerController::OnDeSelect ( )

Blueprint Event, called when de-selecting current unit

3.8.2.9 void ASkirmishPlayerController::OnMove (const FVector & *Location*)

Blueprint Event, called when moving unit

Parameters

Location	we move to

3.8.2.10 void ASkirmishPlayerController::OnOverLocation (const FVector & *Location*)

Blueprint Event, called when hovering over location

Parameters

Location	we hover over

3.8.2.11 void ASkirmishPlayerController::OnOverUnit (ABaseCharacter * *Unit*)

Blueprint Event, called when hovering over unit

Parameters

Unit	we hover over

3.8.2.12 void ASkirmishPlayerController::OnSelect ( )

Blueprint Event, called when selecting new unit

3.8.2.13 void ASkirmishPlayerController::SelectReleased ( )

Released select button, selects or deselects unit

3.8.2.14 void ASkirmishPlayerController::SelectUnit (ABaseCharacter * *NewCharacter*)

Select a new unit

Parameters

NewCharacter	to select

3.8.2.15 void ASkirmishPlayerController::SetTeam (int32 *NewTeam*)

Set the controllers team

Returns

NewTeam to control

3.8.2.16 void ASkirmishPlayerController::ToggleRange ( )

Toggle team range decal visibility for units

3.8.2.17 void ASkirmishPlayerController::ToggleTeamColor ( )

Toggle team color decal visibility for units

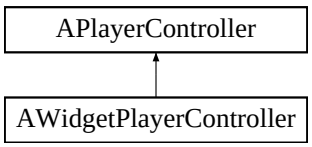
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/PlayerController/Skirmish↩ PlayerController.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/PlayerController/Skirmish↩ PlayerController.cpp

3.9 AWidgetPlayerController Class Reference

`#include <WidgetPlayerController.h>`

Inheritance diagram for AWidgetPlayerController:



Public Member Functions

- virtual void **BeginPlay** () override
- void ChangeUIWidget (TSubclassOf< UUserWidget > NewWidget)

Protected Attributes

- TSubclassOf< UUserWidget > **WUInterface**
- UUserWidget * **PtrUInterface**

3.9.1 Detailed Description

PlayerController used for the main menu.

See also

APlayerController

3.9.2 Member Function Documentation

3.9.2.1 void AWidgetPlayerController::ChangeUIWidget (TSubclassOf< UUserWidget > *NewWidget*)

Change to a different widget

Parameters

NewWidget	to change to.

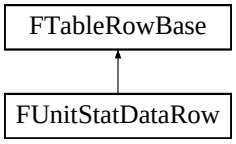
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/PlayerController/Widget↩ PlayerController.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/PlayerController/Widget↩ PlayerController.cpp

3.10 FUnitStatDataRow Struct Reference

`#include <UnitStatDataRow.h>`

Inheritance diagram for FUnitStatDataRow:



Public Attributes

- float **Strength**
- float **Dexterity**
- float **Intelligence**
- float **Constitution**
- float **Speed**
- float **Percepion**
- float **Health**
- float **ActionPoints**
- float **HitChance**
- float **CriticalHitChance**
- float **CriticalHitMultiplier**
- float **Armour**
- float **DodgeChance**
- float **FireResistence**
- float **EarthResistance**
- float **WaterResistance**
- float **AirResistance**
- float **Movement**
- float **Sight**
- float **MaxLevel**
- float **XP**
- float **KilledXp**

3.10.1 Detailed Description

Row for creating data table containing a units statistics

See also

FTableRowBase

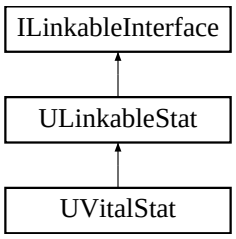
The documentation for this struct was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Structs/UnitStatDataRow.h

3.11 ILinkableInterface Class Reference

`#include <LinkableInterface.h>`

Inheritance diagram for ILinkableInterface:



Public Member Functions

- virtual void AddLinker (UStatLinker ∗Link)
- virtual void RemoveSingleOfLinker (UStatLinker ∗Link)
- virtual void RemoveAllOfLinker (UStatLinker ∗Link)
- virtual void UpdateLinkerValue ()
- virtual void ClearLinkers ()

3.11.1 Detailed Description

Interface for managing linkable statistics

3.11.2 Member Function Documentation

3.11.2.1 void ILinkableInterface::AddLinker (UStatLinker ∗ *Link*) [virtual]

Add link

Parameters

link	to add

Reimplemented in ULinkableStat.

3.11.2.2 void ILinkableInterface::ClearLinkers ( ) [virtual]

Remove all the link

Reimplemented in ULinkableStat.

3.11.2.3 void ILinkableInterface::RemoveAllOfLinker (UStatLinker * *Link*) [virtual]

Remove all of a link

Parameters

link	to remove

Reimplemented in ULinkableStat.

3.11.2.4 void ILinkableInterface::RemoveSingleOfLinker (UStatLinker * *Link*) [virtual]

Remove single of a link

Parameters

link	to remove

Reimplemented in ULinkableStat.

3.11.2.5 void ILinkableInterface::UpdateLinkerValue ( ) [virtual]

Update the linkvalue

Reimplemented in ULinkableStat, and UVitalStat.

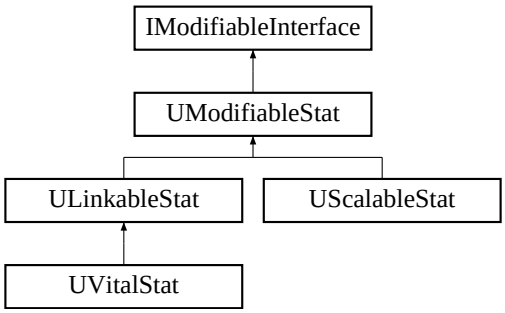
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/Stat↩ Interfaces/LinkableInterface.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/Stat↩ Interfaces/LinkableInterface.cpp

3.12 IModifiableInterface Class Reference

`#include <ModifiableInterface.h>`

Inheritance diagram for IModifiableInterface:



Public Member Functions

- virtual void AddModifier (UStatModifier ∗Modifier)
- virtual void RemoveSingleOfModifier (UStatModifier ∗Modifier)
- virtual void RemoveAllOfModifier (UStatModifier ∗Modifier)
- virtual void UpdateModValue ()
- virtual void ClearModifiers ()

3.12.1 Detailed Description

Interface for managing modifiable statistics

3.12.2 Member Function Documentation

3.12.2.1 void IModifiableInterface::AddModifier (UStatModifier ∗ *Modifier*) [virtual]

Add modifier

Parameters

modifier	to add

Reimplemented in UModifiableStat.

3.12.2.2 void IModifiableInterface::ClearModifiers ( ) [virtual]

Remove all the modifiers

Reimplemented in UModifiableStat.

3.12.2.3 void IModifiableInterface::RemoveAllOfModifier (UStatModifier * *Modifier*) [virtual]

Remove all of a modifier

Parameters

modifier	to remove

Reimplemented in UModifiableStat.

3.12.2.4 void IModifiableInterface::RemoveSingleOfModifier (UStatModifier * *Modifier*) [virtual]

Remove single of a modifier

Parameters

modifier	to remove

Reimplemented in UModifiableStat.

3.12.2.5 void IModifiableInterface::UpdateModValue ( ) [virtual]

Update the modvalue

Reimplemented in UModifiableStat, UVitalStat, ULinkableStat, and UScalableStat.

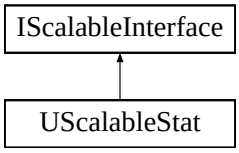
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/Stat↩ Interfaces/ModifiableInterface.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/Stat↩ Interfaces/ModifiableInterface.cpp

3.13 IScalableInterface Class Reference

`#include <ScalableInterface.h>`

Inheritance diagram for IScalableInterface:



Public Member Functions

- virtual void ScaleStat (float Scale)

3.13.1 Detailed Description

Interface for managing scalable statistics

3.13.2 Member Function Documentation

3.13.2.1 void IScalableInterface::ScaleStat (float *Scale*) [virtual]

Sets the scale to a new value

Parameters

scale	to set to

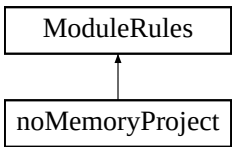
Reimplemented in UScalableStat.

The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatInterfaces/ScalableInterface.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatInterfaces/ScalableInterface.cpp

3.14 noMemoryProject Class Reference

Inheritance diagram for noMemoryProject:



Public Member Functions

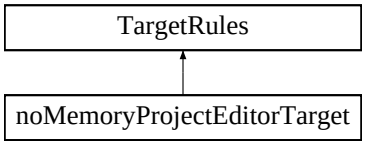
- **noMemoryProject** (TargetInfo Target)

The documentation for this class was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/noMemoryProject.Build.cs

3.15 noMemoryProjectEditorTarget Class Reference

Inheritance diagram for noMemoryProjectEditorTarget:



Public Member Functions

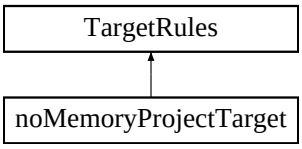
- **noMemoryProjectEditorTarget** (TargetInfo Target)
- override void **SetupBinaries** (TargetInfo Target, ref List< UEBuildBinaryConfiguration > OutBuildBinary↩ Configurations, ref List< string > OutExtraModuleNames)

The documentation for this class was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProjectEditor.Target.cs

3.16 noMemoryProjectTarget Class Reference

Inheritance diagram for noMemoryProjectTarget:



Public Member Functions

- **noMemoryProjectTarget** (TargetInfo Target)
- override void **SetupBinaries** (TargetInfo Target, ref List< UEBuildBinaryConfiguration > OutBuildBinary↩ Configurations, ref List< string > OutExtraModuleNames)

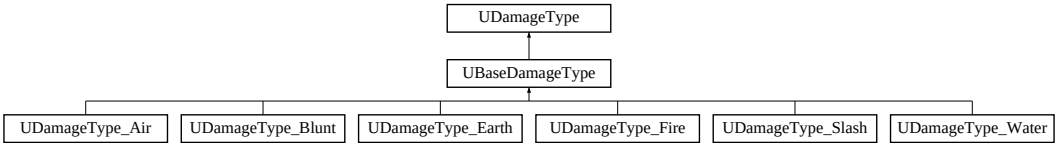
The documentation for this class was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject.Target.cs

3.17 UBaseDamageType Class Reference

`#include <BaseDamageType.h>`

Inheritance diagram for UBaseDamageType:



Public Member Functions

- virtual float CalculateReducedDamage (UStatCollection *StatCollection, float RawDamage)

Public Attributes

- FString **DamageName**

3.17.1 Detailed Description

Base damage type for this game

See also

UDamageType

3.17.2 Member Function Documentation

3.17.2.1 float UBaseDamageType::CalculateReducedDamage (UStatCollection * *StatCollection,* float *RawDamage*) [virtual]

Calculates the damage based on the damage type

Parameters

Character	we are applying the damage type to
RawDamage	before applying damage type

Returns

damage after damage type has been applied

Reimplemented in UDamageType_Air, UDamageType_Blunt, UDamageType_Earth, UDamageType_Fire, UDamageType_Slash, and UDamageType_Water.

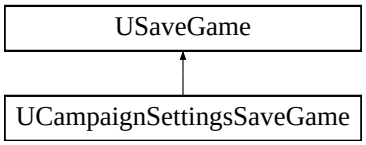
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/DamageTypes/BaseDamageType.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/DamageTypes/BaseDamageType.cpp

3.18 UCampaignSettingsSaveGame Class Reference

`#include <CampaignSettingsSaveGame.h>`

Inheritance diagram for UCampaignSettingsSaveGame:



Public Attributes

- EDifficulty **Difficulty**
- FString **SaveSlotName**
- int32 **UserIndex**

3.18.1 Detailed Description

Class for storing campaign game settings

See also

USaveGame

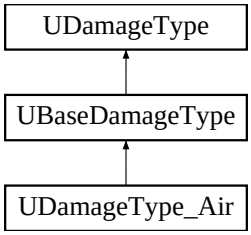
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/SaveGame/Campaign↩ SettingsSaveGame.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/SaveGame/Campaign↩ SettingsSaveGame.cpp

3.19 UDamageType_Air Class Reference

`#include <DamageType_Air.h>`

Inheritance diagram for UDamageType_Air:



Public Member Functions

- virtual float CalculateReducedDamage (UStatCollection ∗StatCollection, float RawDamage) override

Additional Inherited Members

3.19.1 Detailed Description

Air damage type, damage reduced by air resistance

3.19.2 Member Function Documentation

3.19.2.1 float UDamageType_Air::CalculateReducedDamage (UStatCollection * *StatCollection,* float *RawDamage*) [override], [virtual]

See also

UBaseDamageType

Reimplemented from UBaseDamageType.

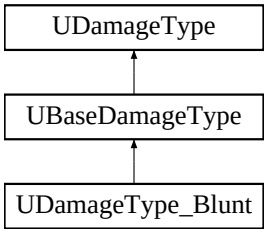
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/DamageTypes/DamageType_Air.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/DamageTypes/DamageType_Air.cpp

3.20 UDamageType_Blunt Class Reference

`#include <DamageType_Blunt.h>`

Inheritance diagram for UDamageType_Blunt:



Public Member Functions

- virtual float CalculateReducedDamage (UStatCollection *StatCollection, float RawDamage) override

Public Attributes

- float **ArmourIgnorePercentage**

3.20.1 Detailed Description

Blunt damage type, damage reduced by armour, ignores some armour

3.20.2 Member Function Documentation

3.20.2.1 float UDamageType_Blunt::CalculateReducedDamage (UStatCollection * *StatCollection,* float *RawDamage*) [override], [virtual]

Calculates the damage based on the damage type

Parameters

Character	we are applying the damage type to
RawDamage	before applying damage type

Returns

damage after damage type has been applied

Reimplemented from UBaseDamageType.

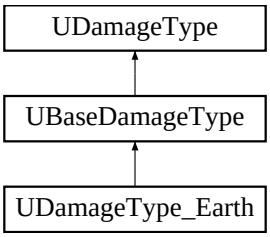
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/DamageTypes/Damage↩ Type_Blunt.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/DamageTypes/Damage↩ Type_Blunt.cpp

3.21 UDamageType_Earth Class Reference

`#include <DamageType_Earth.h>`

Inheritance diagram for UDamageType_Earth:



Public Member Functions

- virtual float CalculateReducedDamage (UStatCollection ∗StatCollection, float RawDamage) override

Additional Inherited Members

3.21.1 Detailed Description

Earth damage type, damage reduced by earth resistance

3.21.2 Member Function Documentation

3.21.2.1 float UDamageType_Earth::CalculateReducedDamage (UStatCollection ∗ *StatCollection,* float *RawDamage*) [override], [virtual]

Calculates the damage based on the damage type

Parameters

Character	we are applying the damage type to
RawDamage	before applying damage type

Returns

damage after damage type has been applied

Reimplemented from UBaseDamageType.

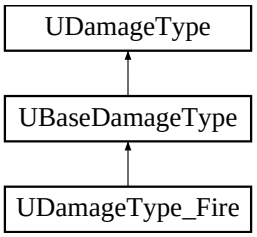
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/DamageTypes/DamageType_Earth.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/DamageTypes/DamageType_Earth.cpp

3.22 UDamageType_Fire Class Reference

`#include <DamageType_Fire.h>`

Inheritance diagram for UDamageType_Fire:



Public Member Functions

- virtual float CalculateReducedDamage (UStatCollection *StatCollection, float RawDamage) override

Additional Inherited Members

3.22.1 Detailed Description

Fire damage type, damage reduced by fire resistance

3.22.2 Member Function Documentation

3.22.2.1 float UDamageType_Fire::CalculateReducedDamage (UStatCollection * *StatCollection*, float *RawDamage*) [override], [virtual]

Calculates the damage based on the damage type

Parameters

Character	we are applying the damage type to
RawDamage	before applying damage type

Returns

damage after damage type has been applied

Reimplemented from UBaseDamageType.

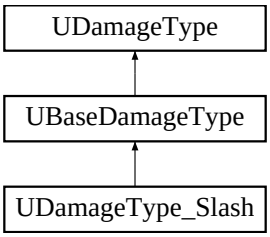
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/DamageTypes/DamageType_Fire.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/DamageTypes/DamageType_Fire.cpp

3.23 UDamageType_Slash Class Reference

`#include <DamageType_Slash.h>`

Inheritance diagram for UDamageType_Slash:



Public Member Functions

- virtual float CalculateReducedDamage (UStatCollection *StatCollection, float RawDamage) override

Additional Inherited Members

3.23.1 Detailed Description

Blunt damage type, damage reduced by armour

3.23.2 Member Function Documentation

3.23.2.1 float UDamageType_Slash::CalculateReducedDamage (UStatCollection * *StatCollection,* float *RawDamage*) [override], [virtual]

Calculates the damage based on the damage type

Parameters

Character	we are applying the damage type to
RawDamage	before applying damage type

Returns

damage after damage type has been applied

Reimplemented from UBaseDamageType.

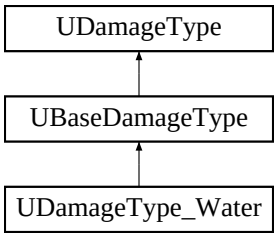
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/DamageTypes/DamageType_Slash.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/DamageTypes/DamageType_Slash.cpp

3.24 UDamageType_Water Class Reference

`#include <DamageType_Water.h>`

Inheritance diagram for UDamageType_Water:



Public Member Functions

- virtual float CalculateReducedDamage (UStatCollection *StatCollection, float RawDamage) override

Additional Inherited Members

3.24.1 Detailed Description

Water damage type, damage reduced by water resistance

3.24.2 Member Function Documentation

3.24.2.1 float UDamageType_Water::CalculateReducedDamage (UStatCollection * *StatCollection,* float *RawDamage*) [override], [virtual]

Calculates the damage based on the damage type

Parameters

Character	we are applying the damage type to
RawDamage	before applying damage type

Returns

damage after damage type has been applied

Reimplemented from UBaseDamageType.

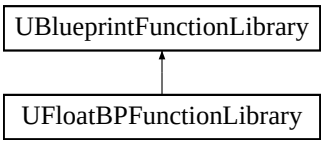
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/DamageTypes/Damage↩ Type_Water.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/DamageTypes/Damage↩ Type_Water.cpp

3.25 UFloatBPFunctionLibrary Class Reference

`#include <FloatBPFunctionLibrary.h>`

Inheritance diagram for UFloatBPFunctionLibrary:



Static Public Member Functions

- static FString GetFloatAsStringWithPrecision (float Float, int32 Precision, bool IncludeLeadingZero=true)

3.25.1 Detailed Description

Utility class for blueprint functions related to floats

See also

UBlueprintFunctionLibrary

3.25.2 Member Function Documentation

3.25.2.1 FString UFloatBPFunctionLibrary::GetFloatAsStringWithPrecision (float *Float,* int32 *Precision,* bool *IncludeLeadingZero* = `true`) `[static]`

Convert float to string with precision

Parameters

Float	to change
Precision	of the float
IncludeLeadingZero	0.5 or .5

Returns

Float as String

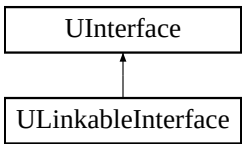
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Utility/FloatBPFunction↩ Library.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/Utility/FloatBPFunction↩ Library.cpp

3.26 ULinkableInterface Class Reference

`#include <LinkableInterface.h>`

Inheritance diagram for ULinkableInterface:



3.26.1 Detailed Description

Class needed for linkable interface in unreal

See also

UInterface

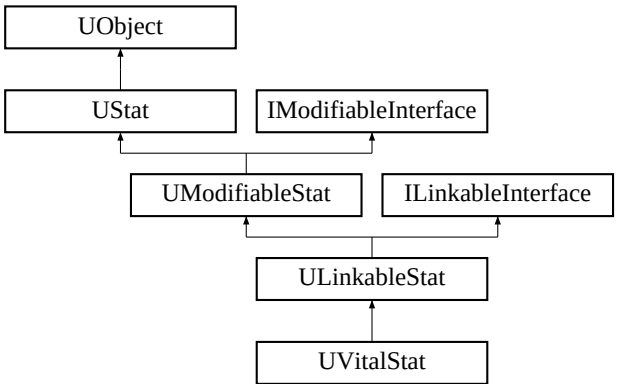
The documentation for this class was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/Stat↩ Interfaces/LinkableInterface.h

3.27 ULinkableStat Class Reference

`#include <LinkableStat.h>`

Inheritance diagram for ULinkableStat:



Public Member Functions

- **ULinkableStat** (const FObjectInitializer &ObjectInitializer)
- void Init (FName Name, float Value)
- float GetLinkValue ()
- virtual float GetFinalValue () override
- virtual void UpdateModValue () override
- virtual void AddLinker (UStatLinker *Link) override
- virtual void RemoveSingleOfLinker (UStatLinker *Link) override
- virtual void RemoveAllOfLinker (UStatLinker *Link) override
- virtual void UpdateLinkerValue () override
- virtual void ClearLinkers () override

Static Public Member Functions

- static ULinkableStat * MakeLinkableStat (FName Name, float Value)

Protected Attributes

- float **LinkValue**
- TArray< UStatLinker * > **Linkers**

3.27.1 Detailed Description

statistic class that allows the final value to derived from other statistics.

See also

UModifiableStat, ILinkableInterface

3.27.2 Member Function Documentation

3.27.2.1 void ULinkableStat::AddLinker (UStatLinker * *Link*) [override], [virtual]

See also

ILinkableInterface

Reimplemented from ILinkableInterface.

3.27.2.2 void ULinkableStat::ClearLinkers ( ) [override], [virtual]

See also

ILinkableInterface

Reimplemented from ILinkableInterface.

3.27.2.3 float ULinkableStat::GetFinalValue ( ) [override], [virtual]

See also

UStat

Reimplemented from UModifiableStat.

3.27.2.4 float ULinkableStat::GetLinkValue ( )

Get the linkvalue of the statistic.

Returns

linkvalue

3.27.2.5 void ULinkableStat::Init (FName *Name,* float *Value*)

Used as a replecement for constructor with parameters

Parameters

Name	of Statistic
Value	for Statistics base

3.27.2.6 ULinkableStat * ULinkableStat::MakeLinkableStat (FName *Name,* float *Value*) [static]

Create a new object with parameters as values

Parameters

name	for statistic
value	for the base value

3.27.2.7 void ULinkableStat::RemoveAllOfLinker (UStatLinker * *Link*) [override], [virtual]

See also

ILinkableInterface

Reimplemented from ILinkableInterface.

3.27.2.8 void ULinkableStat::RemoveSingleOfLinker (UStatLinker * *Link*) [override], [virtual]

See also

ILinkableInterface

Reimplemented from ILinkableInterface.

3.27.2.9 void ULinkableStat::UpdateLinkerValue ( ) [override], [virtual]

See also

ILinkableInterface

Reimplemented from ILinkableInterface.

Reimplemented in UVitalStat.

3.27.2.10 void ULinkableStat::UpdateModValue ( ) [override], [virtual]

See also

UModifiableStat

Reimplemented from UModifiableStat.

Reimplemented in UVitalStat.

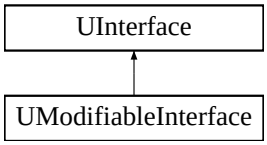
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatTypes/LinkableStat.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatTypes/LinkableStat.cpp

3.28 UModifiableInterface Class Reference

`#include <ModifiableInterface.h>`

Inheritance diagram for UModifiableInterface:



3.28.1 Detailed Description

Class needed for modifiable interfaces in unreal

See also

UInterface

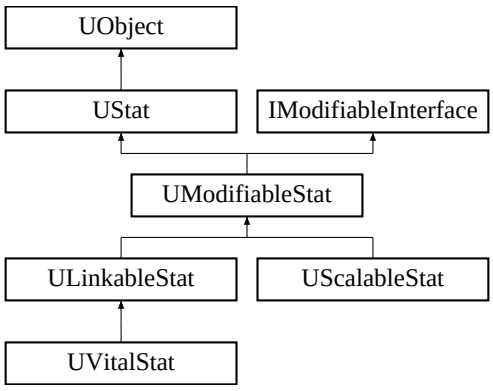
The documentation for this class was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/Stat↩ Interfaces/ModifiableInterface.h

3.29 UModifiableStat Class Reference

`#include <ModifiableStat.h>`

Inheritance diagram for UModifiableStat:



Public Member Functions

- **UModifiableStat** (const FObjectInitializer &ObjectInitializer)
- void Init (FName Name, float Value)
- float GetModValue ()
- virtual float GetFinalValue () override
- virtual void AddModifier (UStatModifier *Modifier) override
- virtual void RemoveSingleOfModifier (UStatModifier *Modifier) override
- virtual void RemoveAllOfModifier (UStatModifier *Modifier) override
- virtual void UpdateModValue () override
- virtual void ClearModifiers () override

Static Public Member Functions

- static UModifiableStat * MakeModifiableStat (FName Name, float Value)

Protected Attributes

- float **ModValue**
- TArray< UStatModifier * > **Modifiers**

3.29.1 Detailed Description

Statistic class that allows modifiers to be added to the final value of the Statistic.

See also

UStat, IModifiableInterface

3.29.2 Member Function Documentation

3.29.2.1 void UModifiableStat::AddModifier (UStatModifier * *Modifier*) [override], [virtual]

See also

IModifiableInterface

Reimplemented from IModifiableInterface.

3.29.2.2 void UModifiableStat::ClearModifiers ( ) [override], [virtual]

See also

IModifiableInterface

Reimplemented from IModifiableInterface.

3.29.2.3 float UModifiableStat::GetFinalValue ( ) [override], [virtual]

See also

UStat

Reimplemented from UStat.

Reimplemented in ULinkableStat, and UScalableStat.

3.29.2.4 float UModifiableStat::GetModValue ( )

Get the modvalue of the Statistic.

Returns

modvalue

3.29.2.5 void UModifiableStat::Init (FName *Name,* float *Value*)

Used as a replecement for constructor with parameters

Parameters

Name	of Statistic
Value	for Statistics base

3.29.2.6 UModifiableStat * UModifiableStat::MakeModifiableStat (FName *Name,* float *Value*) [static]

Create a new object with parameters as values

Parameters

Name	of Statistic
Value	for Statistics base

3.29.2.7 void UModifiableStat::RemoveAllOfModifier (UStatModifier * *Modifier*) [override], [virtual]

See also

IModifiableInterface

Reimplemented from IModifiableInterface.

3.29.2.8 void UModifiableStat::RemoveSingleOfModifier (UStatModifier * *Modifier*) [override], [virtual]

See also

IModifiableInterface

Reimplemented from IModifiableInterface.

3.29.2.9 void UModifiableStat::UpdateModValue ( ) [override], [virtual]

See also

IModifiableInterface

Reimplemented from IModifiableInterface.

Reimplemented in UVitalStat, ULinkableStat, and UScalableStat.

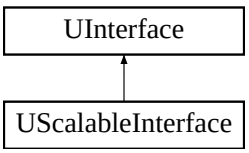
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatTypes/ModifiableStat.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatTypes/ModifiableStat.cpp

3.30 UScalableInterface Class Reference

`#include <ScalableInterface.h>`

Inheritance diagram for UScalableInterface:



3.30.1 Detailed Description

Class needed for scalable interfaces in unreal

See also

UInterface

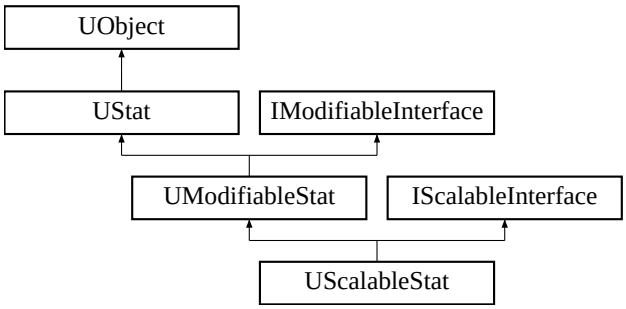
The documentation for this class was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatInterfaces/ScalableInterface.h

3.31 UScalableStat Class Reference

`#include <ScalableStat.h>`

Inheritance diagram for UScalableStat:



Public Member Functions

- **UScalableStat** (const FObjectInitializer &ObjectInitializer)
- void Init (FName Name, float Value, float scValue)
- float GetScaleValue ()
- virtual float GetFinalValue () override
- virtual void UpdateModValue () override
- virtual void ScaleStat (float scale) override

Static Public Member Functions

- static UScalableStat * MakeScalableStat (FName Name, float Value, float scValue=0.0f)

Protected Attributes

- float **ScaleValue**

3.31.1 Detailed Description

statistic class that allows the final value to be scaled.

See also

UModifiableStat, IScalableInterface

3.31.2 Member Function Documentation

3.31.2.1 float UScalableStat::GetFinalValue ( ) [override], [virtual]

See also

UStat

Reimplemented from UModifiableStat.

3.31.2.2 float UScalableStat::GetScaleValue ( )

Get the scale of the statistic.

Returns

modvalue

3.31.2.3 void UScalableStat::Init (FName *Name,* float *Value,* float *scValue*)

Used as a replecement for constructor with parameters

Parameters

Name	of Statistic
Value	for Statistics base
scValue	to set scale value to

3.31.2.4 UScalableStat * UScalableStat::MakeScalableStat (FName *Name,* float *Value,* float *scValue* = `0.0f`) `[static]`

Create a new object with parameters as values

Parameters

Name	of Statistic
Value	for Statistics base
scValue	to set scale value to

3.31.2.5 void UScalableStat::ScaleStat (float *scale*) `[override], [virtual]`

See also

IScalableInterface

Reimplemented from IScalableInterface.

3.31.2.6 void UScalableStat::UpdateModValue ( ) `[override], [virtual]`

See also

UModifiableStat

Reimplemented from UModifiableStat.

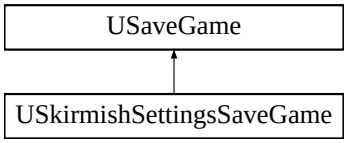
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatTypes/ScalableStat.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatTypes/ScalableStat.cpp

3.32 USkirmishSettingsSaveGame Class Reference

`#include <SkirmishSettingsSaveGame.h>`

Inheritance diagram for USkirmishSettingsSaveGame:



Public Attributes

- EDifficulty **Difficulty**
- FString **SaveSlotName**
- int32 **UserIndex**

3.32.1 Detailed Description

Class for storing skirmish game settings

See also

USaveGame

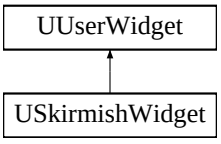
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/SaveGame/Skirmish↩ SettingsSaveGame.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/SaveGame/Skirmish↩ SettingsSaveGame.cpp

3.33 USkirmishWidget Class Reference

`#include <SkirmishWidget.h>`

Inheritance diagram for USkirmishWidget:



Public Member Functions

- void MoveInfo (const float &mouseX, const float &mouseY)
- void ShowHealthBar (ABaseCharacter *Unit)
- void HideHealthBar ()
- void AddToCombatLog (const FString &Line)

3.33.1 Detailed Description

Class that acts as a Base for Skirmish Widget, also enables code to access blueprint events

See also

UUserWidget

3.33.2 Member Function Documentation

3.33.2.1 void USkirmishWidget::AddToCombatLog (const FString & *Line*)

Blueprint Event, Adds string to the combat log

Parameters

Line	to add to the combat log

3.33.2.2 void USkirmishWidget::HideHealthBar ( )

Blueprint Event, Hide the hover over health bar

3.33.2.3 void USkirmishWidget::MoveInfo (const float & *mouseX*, const float & *mouseY*)

Blueprint Event, Move info box displaying cost distance etc

Parameters

mouseX	coordinate
mouseY	coordinate

3.33.2.4 void USkirmishWidget::ShowHealthBar (ABaseCharacter * *Unit*)

Blueprint Event, Show the hover over health bar

Parameters

Unit	that we hover over

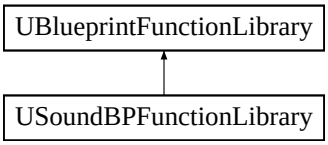
The documentation for this class was generated from the following file:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Widgets/SkirmishWidget.h

3.34 USoundBPFunctionLibrary Class Reference

`#include <SoundBPFunctionLibrary.h>`

Inheritance diagram for USoundBPFunctionLibrary:



Static Public Member Functions

- static bool SetSoundVolume (USoundClass *SoundClassObject, float Volume)
- static float GetSoundVolume (USoundClass *SoundClassObject)

3.34.1 Detailed Description

Utility class for blueprint functions related to sound

See also

UBlueprintFunctionLibrary

3.34.2 Member Function Documentation

3.34.2.1 float USoundBPFunctionLibrary::GetSoundVolume (USoundClass * *SoundClassObject*) [static]

Get volume value from sound class object

Parameters

SoundClassObject	to get from

Returns

volume of sound class

3.34.2.2 bool USoundBPFunctionLibrary::SetSoundVolume (USoundClass * *SoundClassObject,* float *Volume*) [static]

Set the volume for sound class object

Parameters

SoundClassObject	
Volume	to set to

Returns

true if the volume was set

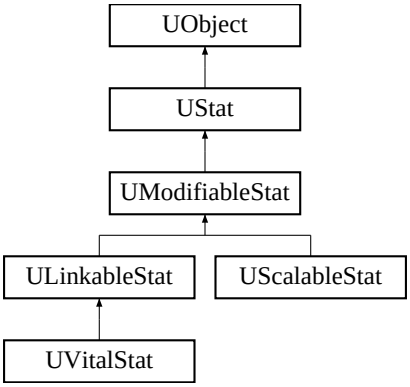
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/Utility/SoundBPFunction↩ Library.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/Utility/SoundBPFunction↩ Library.cpp

3.35 UStat Class Reference

`#include <Stat.h>`

Inheritance diagram for UStat:



Public Member Functions

- **UStat** (const FObjectInitializer &ObjectInitializer)
- void Init (FName Name, float Value)
- FName GetStatName ()
- virtual float GetBaseValue ()
- virtual float GetFinalValue ()
- virtual void SetStatName (FName Name)
- virtual void SetBaseValue (float Value)

Static Public Member Functions

- static UStat * MakeStat (FName Name, float Value)

Protected Attributes

- FName **StatName**
- float **BaseValue**

3.35.1 Detailed Description

Base class for the statistics in the statistics system.

See also

UObject

3.35.2 Member Function Documentation

3.35.2.1 float UStat::GetBaseValue ( ) [virtual]

Get the base value of the statistic.

Returns

base value

3.35.2.2 float UStat::GetFinalValue ( ) [virtual]

Get the value of the statistic with any changes.

Returns

final value

Reimplemented in ULinkableStat, UModifiableStat, and UScalableStat.

3.35.2.3 FName UStat::GetStatName ( )

Get the name of the statistic.

Returns

statistic name

3.35.2.4 void UStat::Init (FName *Name,* float *Value*)

Used as a replecement for constructor with parameters

Parameters

Name	of the statistic
Value	for the base value

3.35.2.5 UStat * UStat::MakeStat (FName *Name,* float *Value*) [static]

Create a new object with parameters as values

Parameters

Name	for statistic
value	for the base value

3.35.2.6 void UStat::SetBaseValue (float *Value*) [virtual]

Set the base value for the statistic.

Parameters

Value	to set to.

3.35.2.7 void UStat::SetStatName (FName *Name*) [virtual]

Set the statistic name for the statistic.

Parameters

Name	to set to.

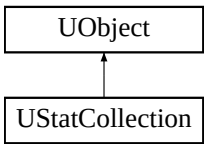
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatTypes/Stat.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatTypes/Stat.cpp

3.36 UStatCollection Class Reference

`#include <StatCollection.h>`

Inheritance diagram for UStatCollection:



Public Member Functions

- virtual bool InitializeMapFromData (UDataTable ∗StatTable, FName RowName)
- void AddStat (EAttribute Type, UStat ∗Stat)
- TMap< EAttribute, UStat ∗ > GetStatMap ()
- bool ContainStat (EAttribute Type)
- UStat ∗ GetStat (EAttribute Type)
- float GetBaseValue (EAttribute Type)
- float GetFinalValue (EAttribute Type)
- float GetFinalValueFloorToFloat (EAttribute Type)
- int32 GetFinalValueFloorToInt (EAttribute Type)
- void SetBaseValue (EAttribute Type, float Value)
- void AddModifierToStat (UStatModifier ∗modifier)
- void RemoveSingleOfModifierInStat (UStatModifier ∗modifier)
- void RemoveAllOfModifierInStat (UStatModifier ∗modifier)
- void UpdateModifiersInStat (EAttribute Type)
- void ClearModifiersInStat (EAttribute Type)
- void AddModifierToAllStats (UStatModifier ∗modifier)
- void RemoveSingleOfModifierInAllStats (UStatModifier ∗modifier)
- void RemoveAllOfModifierInAllStats (UStatModifier ∗modifier)
- void UpdateModifiersInAllStats ()
- void ClearModifiersInAllStat ()
- void ScaleStat (EAttribute Type, float Value)
- void ScaleCollection (float Value)
- float GetCurrentValue (EAttribute Type)
- float GetCurrentValueFloorToFloat (EAttribute Type)
- int32 GetCurrentValueFloorToInt (EAttribute Type)
- float SetCurrentValue (EAttribute Type, float Value)
- float AddToCurrentValue (EAttribute Type, float Value)
- int32 GetSize ()

Protected Attributes

- TMap< EAttribute, UStat ∗ > **Collection**

3.36.1 Detailed Description

Class for handeling a collection of statistics

See also

UObject

3.36.2 Member Function Documentation

3.36.2.1 void UStatCollection::AddModifierToAllStats (UStatModifier ∗ *modifier*)

Add a modifier to all statistics

Parameters

modifier	to add

3.36.2.2 void UStatCollection::AddModifierToStat (UStatModifier * *modifier*)

Add a modifier to the statistic collection

Parameters

modifier	to add

3.36.2.3 void UStatCollection::AddStat (EAttribute *Type*, UStat * *Stat*)

Add statistic to the collection

Parameters

Type	of statistic
statistic	to add

3.36.2.4 float UStatCollection::AddToCurrentValue (EAttribute *Type*, float *Value*)

Add an amount to the current statistic

Parameters

Type	of statistic
Value	to add to current value

Returns

Overflow over max value

3.36.2.5 void UStatCollection::ClearModifiersInAllStat ( )

clear all modvalues in all states

3.36.2.6 void UStatCollection::ClearModifiersInStat (EAttribute *Type*)

Clear all modifiers in the statistic

Parameters

Type	of statistic

3.36.2.7 bool UStatCollection::ContainStat (EAttribute *Type*)

Check if the map contains a certain statistic

Parameters

Type	of statistic

Returns

true if found

3.36.2.8 float UStatCollection::GetBaseValue (EAttribute *Type*)

Gets base value of the statistic

Parameters

Type	of statistic

Returns

base value of that statistic

3.36.2.9 float UStatCollection::GetCurrentValue (EAttribute *Type*)

Get the current value of a statistic

Parameters

Type	of statistic

3.36.2.10 float UStatCollection::GetCurrentValueFloorToFloat (EAttribute *Type*)

Get current value as a floored float

Parameters

Type	of statistic

Returns

current value floored as float

3.36.2.11 int32 UStatCollection::GetCurrentValueFloorToInt (EAttribute *Type*)

Get current value as an int32 allways floored

Parameters

Type	of statistic

Returns

current value floored as int32

3.36.2.12 float UStatCollection::GetFinalValue (EAttribute *Type*)

Gets final value for that statistic

Parameters

Type	of statistic

Returns

final value for that statistic

3.36.2.13 float UStatCollection::GetFinalValueFloorToFloat (EAttribute *Type*)

Get final value as a floored float

Parameters

Type	of statistic

Returns

final value floored to float

3.36.2.14 int32 UStatCollection::GetFinalValueFloorToInt (EAttribute *Type*)

Get final value as a floored int32

Parameters

Type	of statistic

Returns

final value floored to int32

3.36.2.15 int32 UStatCollection::GetSize ( )

Get the amount of statistics in the collection

Returns

Number of statistics

3.36.2.16 UStat * UStatCollection::GetStat (EAttribute *Type*)

Gets a pointer to the statistic

Parameters

Type	of statistic

Returns

pointer to the statistic, null if there is none

3.36.2.17 TMap< EAttribute, UStat * > UStatCollection::GetStatMap ( )

Get the map of statistics

Returns

map of statistics

3.36.2.18 bool UStatCollection::InitializeMapFromData (UDataTable * *StatTable,* FName *RowName*) [virtual]

Initialize the map using a datatable and the row name to use

Parameters

StatTable	containing all statistics
RowName	to use

Returns

true if statistics were filled in

3.36.2.19 void UStatCollection::RemoveAllOfModifierInAllStats (UStatModifier * *modifier*)

remove all instances of a modifier from all statistics

Parameters

modifier	to remove

3.36.2.20 void UStatCollection::RemoveAllOfModifierInStat (UStatModifier * *modifier*)

Remove a all instances of a modifier in the collection

Parameters

modifier	to remove

3.36.2.21 void UStatCollection::RemoveSingleOfModifierInAllStats (UStatModifier * *modifier*)

remove a single instance of a modifier from all statistics

Parameters

modifier	to remove

3.36.2.22 void UStatCollection::RemoveSingleOfModifierInStat (UStatModifier * *modifier*)

Remove a single instance of a modifier in the collection

Parameters

modifier	to remove

3.36.2.23 void UStatCollection::ScaleCollection (float *Value*)

Scale all scalable values to value

Parameters

Value	to set scaler to

3.36.2.24 void UStatCollection::ScaleStat (EAttribute *Type,* float *Value*)

Scale statistic to a value

Parameters

Type	of statistic
Value	to set scaler to

3.36.2.25 void UStatCollection::SetBaseValue (EAttribute *Type,* float *Value*)

Set the base value of a statistic

Parameters

Type	of statistic
value	to set base to

3.36.2.26 float UStatCollection::SetCurrentValue (EAttribute *Type,* float *Value*)

Set current value of a statistic

Parameters

Type	of statistic
Value	to set current value to

3.36.2.27 void UStatCollection::UpdateModifiersInAllStats ( )

update modvalue in all statistics

3.36.2.28 void UStatCollection::UpdateModifiersInStat (EAttribute *Type*)

Update the modvalue in the statistic

Parameters

Type	of statistic

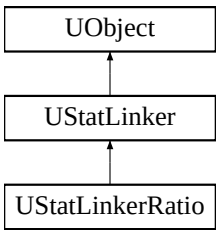
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatCollection.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatCollection.cpp

3.37 UStatLinker Class Reference

`#include <StatLinker.h>`

Inheritance diagram for UStatLinker:



Public Member Functions

- UStat * GetStat ()
- virtual float GetValue ()

Protected Attributes

- UStat * **Stat**

3.37.1 Detailed Description

Wrapper class for a statistic used for linking.

See also

UObject

3.37.2 Member Function Documentation

3.37.2.1 UStat * UStatLinker::GetStat ( )

Get the statistic wrapped in the class.

Returns

statistic pointer

3.37.2.2 float UStatLinker::GetValue ( ) [virtual]

Get the value of the statistic wrapped in the class.

Returns

final value of the statistic

Reimplemented in UStatLinkerRatio.

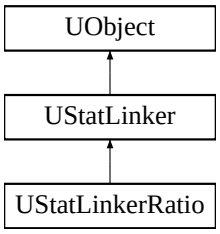
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatLinkers/StatLinker.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatLinkers/StatLinker.cpp

3.38 UStatLinkerRatio Class Reference

`#include <StatLinkerRatio.h>`

Inheritance diagram for UStatLinkerRatio:



Public Member Functions

- **UStatLinkerRatio** (const FObjectInitializer &ObjectInitializer)
- void Init (UStat *LinkStat, float LinkRatio)
- virtual float GetValue () override

Static Public Member Functions

- static UStatLinkerRatio * MakeStatLinker (UStat *LinkStat, float LinkRatio)

Additional Inherited Members

3.38.1 Detailed Description

Class that applies a ratio to the final linkable value.

See also

UStatLinker

3.38.2 Member Function Documentation

3.38.2.1 float UStatLinkerRatio::GetValue ( ) [override], [virtual]

Get the value of the statistic wrapped in the class.

Returns

final value of the statistic times ratio

See also

UStatLinker

Reimplemented from UStatLinker.

3.38.2.2 void UStatLinkerRatio::Init (UStat * *LinkStat*, float *LinkRatio*)

Used as a replecement for constructor with parameters

Parameters

LinkStat	to use as linker
LinkRatio	to use

3.38.2.3 UStatLinkerRatio * UStatLinkerRatio::MakeStatLinker (UStat * *LinkStat*, float *LinkRatio*) [static]

Create a new object with parameters as values

Parameters

LinkStat	to use as linker
LinkRatio	to use

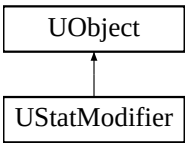
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatLinkers/StatLinkerRatio.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatLinkers/StatLinkerRatio.cpp

3.39 UStatModifier Class Reference

`#include <StatModifier.h>`

Inheritance diagram for UStatModifier:



Public Member Functions

- EAttribute GetAttribute ()
- EModifierType GetModifier ()
- float GetValue ()
- void SetAttribute (EAttribute NewAttribute)
- void SetModifier (EModifierType NewModifier)
- void SetValue (float ModifierValue)

3.39.1 Detailed Description

Class containing value and operation type for a modifier

See also

UObject

3.39.2 Member Function Documentation

3.39.2.1 EAttribute UStatModifier::GetAttribute ( )

Get attribute this modifier is applied to

Returns

attribute type

3.39.2.2 EModifierType UStatModifier::GetModifier ( )

Get operation type of the modifier.

Returns

modifier type

3.39.2.3 float UStatModifier::GetValue ( )

Get the value of the modifier.

Returns

modifier value

3.39.2.4 void UStatModifier::SetAttribute (EAttribute *NewAttribute*)

Set the attribute type to a new one.

Parameters

NewAttribute	to change to

3.39.2.5 void UStatModifier::SetModifier (EModifierType *NewModifier*)

Set the modifier type to a new one.

Parameters

NewModifier	to change to

3.39.2.6 void UStatModifier::SetValue (float *ModifierValue*)

Set the value of the modifier to a new one.

Parameters

ModifierValue	to set modifier value to

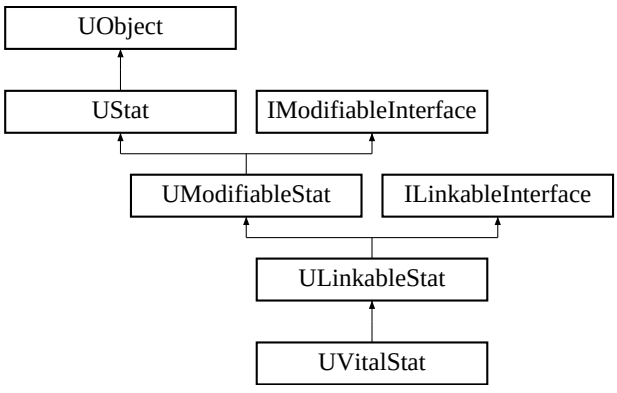
The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/Stat↩ Modifiers/StatModifier.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/Stat↩ Modifiers/StatModifier.cpp

3.40 UVitalStat Class Reference

`#include <VitalStat.h>`

Inheritance diagram for UVitalStat:



Public Member Functions

- **UVitalStat** (const FObjectInitializer &ObjectInitializer)
- void Init (FName Name, float Value, float crntValue)
- float GetCurrentValue ()
- float SetCurrentValue (float Value)
- float AddToCurrentValue (float Value)
- void SetCurrentValueToMax ()
- virtual void UpdateModValue () override
- virtual void UpdateLinkerValue () override

Static Public Member Functions

- static UVitalStat * MakeVitalStat (FName Name, float Value, float crntValue=0.0f)

Protected Attributes

- float **CurrentValue**

3.40.1 Detailed Description

statistic class that has a current value in the range of final value and 0.

See also

ULinkableStat

3.40.2 Member Function Documentation

3.40.2.1 float UVitalStat::AddToCurrentValue (float *Value*)

Add to the current value of the statistic.

Parameters

value	to add

Returns

the amount overflowing the final value

3.40.2.2 float UVitalStat::GetCurrentValue ( )

Get the Currentvalue of the statistic.

Returns

Currentvalue

3.40.2.3 void UVitalStat::Init (FName *Name,* float *Value,* float *crntValue*)

Used as a replecement for constructor with parameters

Parameters

Name	of statistic
Value	for statistics base
crntValue	is the current value of the statistic,

3.40.2.4 UVitalStat * UVitalStat::MakeVitalStat (FName *Name,* float *Value,* float *crntValue* = `0.0f`) `[static]`

Create a new object with parameters as values

Parameters

Name	of Statistic
Value	for Statistics base
crntValue	is the current value of the statistic, default 0.0

3.40.2.5 float UVitalStat::SetCurrentValue (float *Value*)

Set the current value of the statistic.

Parameters

value	to set to

Returns

the amount overflowing the final value

3.40.2.6 void UVitalStat::SetCurrentValueToMax ( )

Sets the current value to the max possible value.

3.40.2.7 void UVitalStat::UpdateLinkerValue ( ) `[override],[virtual]`

See also

ULinkableStat

Reimplemented from ULinkableStat.

3.40.2.8 void UVitalStat::UpdateModValue ( ) [override], [virtual]

See also

UModifiableStat

Reimplemented from ULinkableStat.

The documentation for this class was generated from the following files:

- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Public/StatisticsSystem/StatTypes/VitalStat.h
- E:/Repositories/noMemory/noMemoryProject/Source/noMemoryProject/Private/StatisticsSystem/StatTypes/VitalStat.cpp

Index