

**UNIVERSITAT
JAUME·I**

Synthesis, conceptualization and development of real-time strategy game prototype for mobile and PC platforms

Dominik Konrad Kwiek

Final Degree Work
Bachelor's Degree in
Video Game Design and Development
Universitat Jaume I
June 21, 2021

Supervised by: Sven Casteleyn

Acknowledgments

I want to thank to tutor Sven Casteleyn who allowed and supported me to create this crazy and overly ambitious project and to my friends and family, who have always believed in me, not only during the development of the project but also during all the 4 years of the degree, reminding me that I am capable of whatever it takes to carry on even though in some moments I was about to lose hope.

Abstract

This project is an elaboration of a real-time mobile strategy video game that consists of a simplification of the classics of real-time strategy (RTS) genre from PC platforms through a hexagonal grid, a simplification of the resource system and a dynamic user interface (UI) system. The resource system includes troops that implement artificial intelligence for their behaviors and act inside the hexagonal grid. This behavior is separated into two areas: individual units that form part of groups, and groups themselves which are assigned to one hexagonal grid. The resources system also includes a building system that allows placing structures in the grid. The grid only appears in the areas where the player can interact with.

Contents

1 Introduction	6
1.1 Work Motivation	6
1.2 Objectives	7
1.3 Environment and Initial State	8
2 Planning and resources evaluation	8
2.1 Planning	8
2.2 Resource Evaluation	11
3 Game Design Document	12
3.1 Game Overview	12
3.1.1 Game Concept	12
3.1.2 Feature Set	12
3.1.3 Genre	12
3.1.4 Target Audience	12
3.1.5 Game Flow Summary	13
3.1.6 Look and Feel	13
3.1.7 Project Scope	13
3.2 Gameplay and Mechanics	14
3.2.1 Gameplay	14
3.2.1.1 Actions	14
3.2.1.1.1 Move or attack troop	14
3.2.1.1.2 Construct buildings	14
3.2.1.1.3 Throw spells (optional implementation)	14
3.2.1.1.4 Building actions	14
3.2.1.1.5 Create a troop	14
3.2.1.1.6 Set created troop destination	14
3.2.1.1.7 Assign workers	14
3.2.1.1.8 Deassign workers	14
3.2.1.1.9 Demolish	14
3.2.1.2 Game Progression	14
3.2.1.3 Main levels structure	15
3.2.1.4 Objectives	15
3.2.1.5 Play Flow	15
3.2.2 Mechanics	15
3.2.2.1 Resources	15
3.2.2.2 Troops	16
3.2.2.2.1 Attributes	16
3.2.2.2.2 Healing	17
3.2.2.2.3 Troop types	17
3.2.2.3 Buildings	17
3.2.2.3.1 Attributes	18
3.2.2.3.2 Types	18

3.2.2.3.3 Healing	19
3.2.2.4 Combat	19
3.2.3 Screenflow	20
3.2.3.1 Screen Flow Chart	20
3.2.3.2 Screen Descriptions	20
3.2.4 Game Options	21
3.3 Story, Setting and Character	21
3.3.1 Story and Narrative	21
3.3.1.1 Back story	21
3.3.1.2 Plot Elements	21
3.3.2 Game World	21
3.3.2.1 General look and feel of world	21
3.3.2.2 Factions	22
3.3.2.2.1 Lords of the ashes	22
3.3.2.2.2 Followers of Moses	22
3.4 Levels	23
3.4.1 Level #1	23
3.4.1.1 Synopsis	23
3.4.1.2 Introductory Material	23
3.4.1.3 Objectives	23
3.4.1.4 Physical Description	23
3.5 Interface	24
3.5.1 Visual System	24
3.5.2 Control System	25
3.5.3 Audio	25
3.5.3.1 Music	25
3.5.4 Sound Effects	26
3.6 Artificial Intelligence	26
3.6.1 Opponent AI	26
3.6.2 Support AI	26
3.6.2.1 Player and Collision Detection	26
3.6.2.2 Pathfinding	26
4 System Analysis and Design	26
4.1 Requirement Analysis	26
4.1.1 Functional Requirements	28
4.1.2 Non-functional Requirements	28
4.2 System Design	29
4.3 System Architecture	35
4.3.1 Android	35
4.3.2 Desktop	36
4.4 Interface Design	36
5 Work Development and Results	37
5.1 Work Development	37

5.1.1 Game design document	37
5.1.2 Setting up the project and the Github problem	37
5.1.3 Hex grid	38
5.1.4 Special terrain types and resources	39
5.1.5 Troops and units	39
5.1.5.1 Base development	39
5.1.5.2 Troop attributes	40
5.1.5.3 Troops behaviour	41
5.1.5.4 Units behaviour	42
5.1.5.5 Types of troops	44
5.1.6 Buildings	46
5.1.6.1 Attributes	46
5.1.6.2 Buildings behaviour	47
5.1.6.3 Types of buildings	47
5.1.7 Dynamic User Interface	54
5.1.8 Level Manager	54
5.1.9 Double interaction buttons	55
5.1.10 Camera movement	55
5.1.11 Enemy Artificial Intelligence	55
5.1.12 Game Screens	56
5.1.13 First Level	56
5.1.14 Sound	57
5.2 Results	57
6 Conclusions and Future Work	58
6.1 Conclusions	59
6.2 Future work	59
7 Bibliography	59
8 Other considerations	60
8.1 Source Code	60
8.2 List of Figures and tables	60

1 Introduction

This section presents the motivations of the project and the environment in which it was created.

1.1 Work Motivation

There have been many different approaches to mobile strategy games, from “Clash of clans” [19] to “Bad North” [20] , “Siege Up!” [21] and “Stormbound” [22] among many others, both from real-time and turn-based perspectives. However, the number of games that create a similar experience of a classic RTS PC platform game is not so large. These games usually tend to focus too much on particular aspects of strategy games and simplify others. For example, “Clash of Clans” reduces the number of resources and takes back most of the player’s control over his/her units, by only letting control over where the units are placed, and focus on the player versus player aspect. There is also another case scenario where the game makes the gameplay more difficult because of the excess of options for a mobile interface. To give an example, in “RTS Siege Up! - Medieval Warfare Strategy Offline” the player is given so many options to the point that most of the effort that the player has to make in order to play the game is in handling its interface.

This project attempts to provide a similar experience to that of a classic RTS from PC in the mobile platform by synthesizing the elements of which it is composed while sharpening the UI and creating a universe that will immerse the player through its world and soundtrack. For this purpose, the video game will be implemented in a hexagonal matrix that will ease the selection of troops, one troop for every hexagon, and the building placement. It will also simplify the resource recollection by moving its responsibility from directly the workers to specific buildings. The use of the UI is facilitated by a dynamic UI system that pops up different windows on the screen depending on the element selected by the player in the game. These windows show specific functionalities of the selected element.

The creation of a universe world is accelerated by the use of Synth Store modular assets and Rexards icons. Modular assets are 3D models that fit together to create 3D game environments and objects.



Figure 1.1: Screenshot of 'Bad North' [20] gameplay



Figure 1.2: Screenshot of 'Siege Up!' [21] trailer

1.2 Objectives

The objectives of this project are strictly related to the elaboration of the prototype:

- Implementation of full 2 levels of artificial intelligence for the control of troops.
- Implementation of a dynamic, easy to use and deep user interface.
- Elaboration of a building system.
- Adaptation of external 3D models and sound assets to an individual project while minimizing the loss of originality.
- Development of different types of units and buildings by using generic classes.
- Implementation of the first level.

- Development of a polished prototype that shows the potential of the game mechanics and concept

1.3 Environment and Initial State

The development of the project started in February. The working plan was to dedicate between 4-6 hours to the project from Tuesday to Saturday until May, the month I started my internship, with some days off. With this plan, I archived that, by the start of May, around 240 hours of work were dedicated to the project. I went back to work on the project around the middle of May, raising the I dedicated to the project until I managed around 14-16 hours per week.

In which it refers to my personal capacities, I have spent enough time with unity to feel comfortable using it, but I have never built an RTS before, which has meant that I have been overwhelmed by all the different systems and the relations between them that this genre requires.

2 Planning and resources evaluation

This chapter shows the planning and the development of the project.

2.1 Planning

In the next table, I will illustrate the task developed in order to complete the project. Some of the tasks from the technical proposal have been removed, as they are nonessential to the creation of the prototype or as the logic of the program has changed. Those tasks are:

- Create extension for troops class for melee unit groups and create extension for troops class for range unit groups: After I realised that the behavior of the melee troops and the ranged troops was nearly identical, as they both move to until they are in range to perform an assault to the enemy troop, and then order its units to launch attacks to the enemy units. The difference between these troops is that when the ranged troop is further from one hex cell of distance, when the troops performs an attack to the enemy troop it does not receive damage and also its units do not need to move close to the enemy units.
- Elaborate alert system of user interface: The alert system was indicated as optional and, due to scale of the resto elements of the project, was not implemented.
- Design and search or creation of sound: Due to lack of time, I could not perform this task for the submission.
- Elaborate development logs videos: This element of the project was not indicated as optional. I was aware that an idea cannot go very far if it is not correctly promoted. However, during the development of the game I realized how ambitious this project was, and I took the decision of putting aside the development logs videos.

Also, some new tasks have been added as it was expected that, due to the complexity of the project, not all the tasks could be planned from the start.

Tasks	Time dedicated
Create a hexagonal grid data structure	4
Elaborate generic class for troops	52
Develop troop movement	6
Develop mele unit behavior	22
Develop range unit behavior	9
Search or create units animations	7
Elaborate base user interface	8
Implement building interface system	9
Create troop status interface	5
Create GDD	57
Design, implementation and adaptation of the first level	6
Design, implementation and adaptation of player and enemy buildings	22
Design, implementation and adaptation of player and enemy troops	10
Elaborate memory	53
Polish and additional elements (magic system, day/night system,...)	15
Solving problems with Unity's GitHub asset	12
Investigate how Unity's "New input system" works and implement it in the game	6
Elaborate Hexagonal User Interface	11
Develop a level manager	27
Develop a camera manager	3
Develop enemy AI	8
Total	352

Table 2.1. Tasks and time dedicated

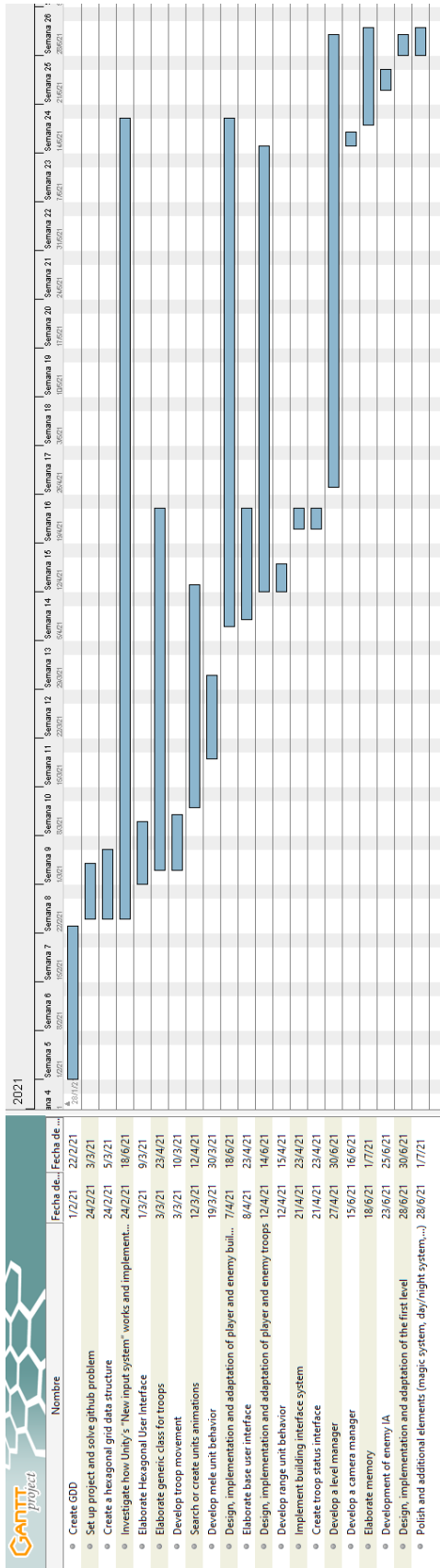


Figure 2.1: Gantt chart of the tasks (made with Gantt Project [18])

2.2 Resource Evaluation

The resources I used to develop this project are:

- Hardware:
 - A secondhand laptop with the following specifications:
 - Operating System: Windows 10 Home 64-bit (10.0, Build 19041) (19041.vb_release.191206-1406)
 - System Manufacturer: ASUSTeK COMPUTER INC.
 - System Model: G751JY
 - BIOS: G751JY.211 (type: UEFI)
 - Processor: Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz (8 CPUs), ~2.6GHz
 - Memory: 16384 MB RAM
 - Available OS Memory: 16334 MB RAM
 - DirectX Version: DirectX 12
 - Cost: 750 €
 - Xiaomi Pocophone F1 device with android operating system. Cost: 300€
- Software:
 - Unity 3D engine 2020.2.6f1 [3]: This is the engine of the game. Cost: 0€.
 - Model and Texture List. These are from the “Humble best of polygon game dev bundle!” bought from a limited time offer at the Humble bundle [23] store. Cost: 18€.
 - Synty Store POLYGON - Knights Pack [12]
 - Synty Store POLYGON - Adventure Pack [13]
 - Synty Store POLYGON - Vikings Pack [14]
 - Animation List
 - Mixamo library [9]: This is a library from Adobe that allows to download animations and models for projects. Cost: 0€
 - Effects List
 - Synty Store POLYGON - Particle FX Pack (Unity Only) [15]. Cost: 35.72€. However, the price can be reduced with coupons and deals to around 10€.
 - Interface Art List
 - Icons by Rexard [16]. Those icons are from a bundle sold for a limited time from the Humble Bundle store. Concretely, the “Humble 7000 game dev icons bundle!” bundle. Cost: 18,5€.
 - Writing documentation applications
 - Google docs [2]. Cost: 0€
 - Libreoffice [8]. Cost: 0€
 - Grammarly [11]. Cost: 0€
 - Other. These assets were bought as part of the “Humble unity games and game dev assets bundle!” limited offer at the Humble bundle store. Cost: 24.54€
 - Unity Peek - Editor Toolkit [17]

The assets that do not appear in the list but are indicated in the Asset List of the GDD were not used in the project. Those are the “Synty Store POLYGON - Samurai Pack” [24], the specified sound assets and “Unity Databox - Data editor & save solution” [25].

3 Game Design Document

I have considered it important to add the main elements of the game design document, in order to explain in more detail some elements of the design of the project and also to have a point of comparison between the original design and the final design. A good part of the optional features from the original document has been removed, as it did not give relevant information for the prototype and game idea.

3.1 Game Overview

3.1.1 Game Concept

This project consists on a real time mobile strategy video game that consists of a simplification of the classics of real time strategy (RTS) genre from PC platforms through a hexagonal grid, a simplification of the resource system and a dynamic user interface (UI) system. The resource system will include troops that will implement artificial intelligence for their behaviours and will act inside the hexagonal grid. This behaviour is separated into two areas: individual units that form part of groups, and groups themselves which are assigned to one hexagonal grid. The resources system also includes a building system that allows to place structures in designated places or in most places of the grid depending on the type of building. The development is also exposed and documented through a video development log format on YouTube.

3.1.2 Feature Set

- Hexagonal grid database
- Resource system
- Troop support IA
- Opponent IA
- Dynamic user interface
- Building system
- Immersive world and narrative
- Soundtrack
- Development log

3.1.3 Genre

Single player mobile RTS

3.1.4 Target Audience

23 years old adult males with an 8 years of age range variation from the target age.

3.1.5 Game Flow Summary

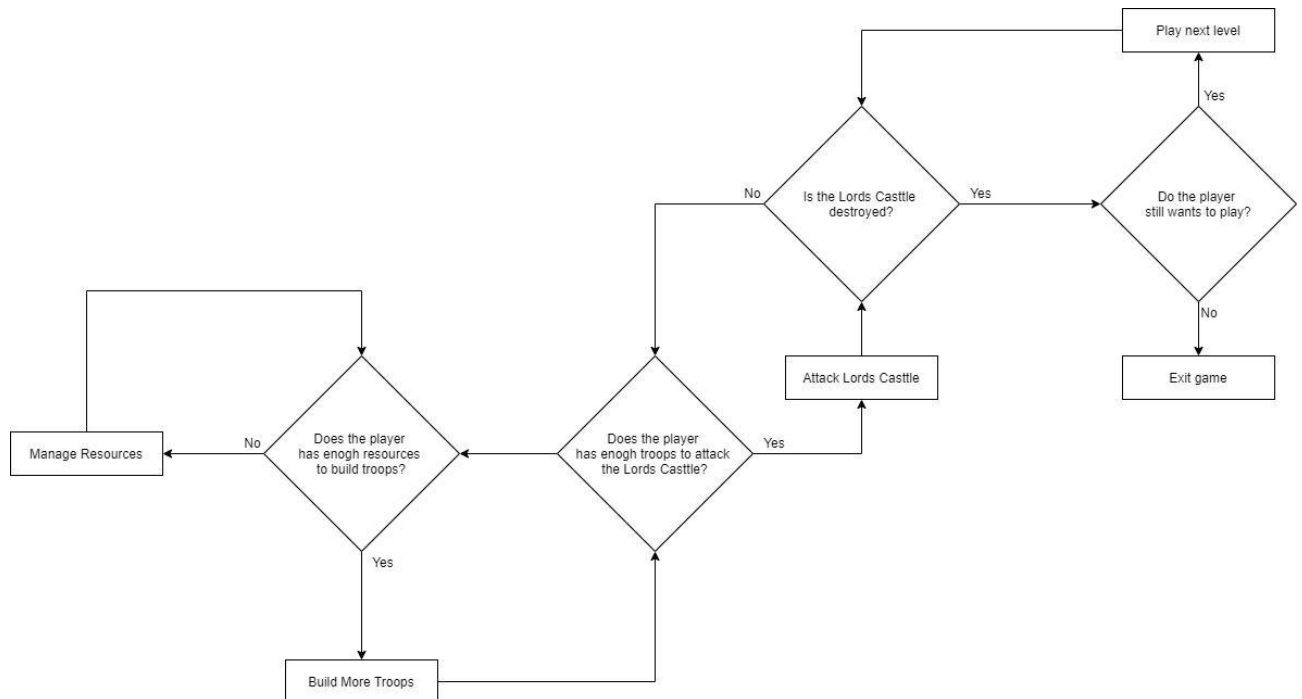


Figure 3.1: Game flow chart summary

3.1.6 Look and Feel

The game is ambient in a medieval dark fantasy inspired by the Warhammer fantasy setting. The player side is influenced by Vikings culture and the enemy side is influenced by medieval English kingdoms.

The visual style is a low poly one, specifically the one used by “Synty Store” in its “polygon” style, allowing better performance for mobile platforms and giving a layer of abstraction.

3.1.7 Project Scope

Number of levels: Minimum of one

Number of type of troops: Minimum of 5

Number of classes of troops: Minimum of 2

Number of type of buildings: Minimum of 13

Number of UI base elements: 4

Number of development video logs: Between 2 and 4.

3.2 Gameplay and Mechanics

3.2.1 Gameplay

3.2.1.1 Actions

3.2.1.1.1 Move or attack troop

The player can select and move a selected troop from a specific hexagon to another empty hexagon. If the hexagon isn't empty but there is an enemy troop or building in it, the selected troop will move to attack the troop or building.

3.2.1.1.2 Construct buildings

The player can select a hexagon and construct a building if the player has the needed resources and the hexagon fulfils the requirements for the building.

3.2.1.1.3 Throw spells (optional implementation)

A variety of spells appears on the middle-down user interface that can be through using Ornus. Those spells have different effects over the troops, resources and game world.

3.2.1.1.4 Building actions

Each building can have a set of actions that can be executed by the player if he/she has the needed resources. This are separated in:

3.2.1.1.5 Create a troop

Creates a troop on a near road to the building. If all nearby roads are occupied, will ask one of the troops to move. If it is not possible, it won't generate the troop and will wait. All created troops start with the "advance and attack" attribute activated.

3.2.1.1.6 Set created troop destination

Set a destination to move to all new troops created.

3.2.1.1.7 Assign workers

Assign one or multiple workers to the building. It will not assign more workers if the maximum is reached.

3.2.1.1.8 Deassign workers

Deassign one or multiple workers from the building. It will not deassign more workers if the total of workers is 0.

3.2.1.1.9 Demolish

Destroy a building and leave the hexagon empty.

3.2.1.2 Game Progression

The players begin every level with a set of buildings, troops and resources in a level where most of the terrain is controlled or in some way protected by the enemy. In order to destroy

the main enemy base and complete the level, the player has to take control and use the enemy resources to develop an army that can be a match against the enemy base garrison, and at the same time defend itself from enemy attacks that come in waves. Every completed level unlocks new units, buildings, spells and/or resources.

3.2.1.3 Main levels structure

Each level has one main enemy base, the objective to destroy in the level, and one or multiple outposts, those are bases with a reduced size that, as the main base, will spawn enemies. In order to destroy the main base, the player will have to destroy at least one outpost.

Each level has resources outside the main enemy base, outpost and player that can have a garrison that has to be defeated.

3.2.1.4 Objectives

The main objective is to destroy the enemy's main base and progress through the story.

The side objectives are constructing buildings, collecting resources, creating troops, exploring and destroying enemy troops, buildings and outposts.

3.2.1.5 Play Flow

At the beginning of each level, the enemy will not send waves of troops to the player base and he/she will be free to explore the terrain and, if he/she is able to, expand his/her territory. As time goes by, waves of enemies will appear more frequently and will become stronger, making the game more complex and intense.

The game will get harder each level.

Optional feature

The world has a day/night cycle, where in the day the enemy waves are weaker or not existent and the player has more vision, making it a better moment to gather resources, expand territory, build and explore. On the other side, the night is the most dangerous moment, where the stronger enemy waves will appear, making it the most combat and defence focused part.

3.2.2 Mechanics

3.2.2.1 Resources

All resources can be collected in special buildings.

Optional feature

Resources can also be looted.

Wood

Basic resources needed to create buildings and some troops. It is collected in the sawmill.

Settlers

Workers that the player needs to use buildings and create troops. They can be collected in settlement centres or ports.

Optional feature

Settlers can be claimed if founded in the game world.

Settlers need shacks to be kept.

Food

Resource needed to maintain troops and settlers. Without food, units from troops and settlers will go away. It can be collected in ports, farms, hatchery and hunting shacks.

Iron

Resource needed to build average troops. It can be collected in iron mines.

Moon silver(Optional feature)

Resource needed to build advanced troops. It can be collected in moon silver mines.

Gold(Optional feature)

Resource needed to build and maintain advanced troops. It can be collected in gold mines, markets and ports.

Ornus(Optional feature)

Resource needed to use spells. It can be collected in herbals.

Stone(Optional feature)

Resource needed to build average and advanced buildings.

3.2.2.2 Troops

Collection of units of the same type that belongs to a faction and is established in a hexagon. All troops are generated in settlement centres except if they belong to the IA side, then they are generated in spawn buildings.

3.2.2.2.1 Attributes

Hit-points(HP)

Numerical value that describes the status of the troop. Have a direct relation with the units of the troop. If it reaches 0, the troop is dispersed. Each troop has its own maximum.

Units

Numerical value that describes the total units of the troops. Each troop has its own maximum.

Attack

Range of hit-points that can a troop take in an attack from an enemy troop. It depends on the total of units.

Defence

Range of hit-points that a troop can save from a melee or other non range type of attack. It depends on the total of units.

Range Defence (Optional feature)

Range of hit-points that a troop can save from a range attack. It depends on the total of units.

Charge attack (Optional feature)

Range of hit-points that a troop can take in an attack from an enemy troop in its first attack. It depends on the total of units. It has a could-down.

Range

Range of hexagons within which the trooper is able to attack.

Movement speed

Hexagons that a troop is able to move in a span of time.

Food maintain

Food that consumes a troop every day or in a determined span of time.

Gold maintain (Optional feature)

Gold that consumes a troop every day or in a determined span of time.

Faction

Faction to which belongs the troop.

Cost for spawn

Resources needed in order to spawn the troop.

Advance and attack

If activated, it will move to attack any troop near even when moving. It can be toggled on or off in the top-middle interface.

If deactivated, the troop only attacks enemy troops in attack range.

Attack frequency

Times a troop attack in a space of time.

3.2.2.2 Healing

Troops heal over time until they reach their maximum Hit-points for their actual value of units.

3.2.2.3 Troop types

Thrasher

Basic troop. Low-average attack damage, low defence, low range defence, high HP, high total unit maximum, high charge attack, melee range, average movement speed, low food maintain, no gold maintain and average attack frequency. Needs wood and settlers to be created.

Hunter

Basic troop. Low-average attack damage, very low defence, very low range defence, medium HP, average total unit maximum, no charge attack, range attacks, average movement speed, low food maintain, no gold maintain and low-average attack frequency. Needs wood and settlers to be created.

Swordsman

Average troop. Average attack damage, medium defence, low-medium range defence, medium HP, medium total unit maximum, medium charge attack, melee range, medium movement speed, average food maintain, no gold maintain and average attack frequency. Needs iron, a forge in use and settlers to be created.

Zombie

Exclusive IA basic troop. Low attack damage, low defence, low range defence, very high HP, very high total unit maximum, low charge attack, melee range, slow-medium movement speed, no food maintain, no gold maintain and average attack frequency.

Knight

Exclusive IA advanced troop. High attack damage, very high defence, high range defence, average HP, low total unit maximum, average-high charge attack, melee range, slow-medium movement speed, no food maintain no gold maintain and slow attack frequency.

3.2.2.3 Buildings

Buildings make hexagons non walkable, except if it is a road.

3.2.2.3.1 Attributes

HP

Numerical value that describes the status of the building. If it reaches 0, the building is destroyed and makes the hexagon unoccupied. Each building has its own maximum.

Workers

Total of settlers assigned to the building. It has a direct relation with production. Each building has its own maximum.

Production

Total of resources produced by a building.

General defence

Range of hit-points that a building can save from an attack.

Cost of construction

Resources needed in order to create the building.

3.2.2.3.2 Types

Settlers Center

Collects settlers and a small amount of wood. Settlers tend to build their shacks nearby. If all settlers centres are destroyed the player loses. It can create units and can not be demolished. The cost of building it elevates for every settlers centre in the player side. High cost of construction. High HP and general defence.

Port

Collects settlers and produces food and gold. It can have assigned and deassigned workers. If demolished, all workers become unemployed. High cost of construction. Average HP and weak general defence. Needs to be built on a water hexagon.

Sawmill

Produces wood. It can have assigned and deassigned workers. Low cost of construction. Average HP and general defence. If demolished, all workers become unemployed.

Hunting shack

Produces food. It can have assigned and deassigned workers. Average cost of construction. Average HP and general defence. If demolished, all workers become unemployed.

Iron mines

Collects iron. It can have assigned and deassigned workers. Need iron nearby. Average cost of construction. Average HP and general defence. If demolished, all workers become unemployed.

Shack (Optional feature)

Home of settlers. Each shack has its maximum of settlers. Shacks are automatically built around Settlers Centres in empty hexagons. It is possible to build over shacks, but it will demolish them. If a shack is demolished, the player will lose the settlers that habit it.

Forge

Needed for production of specific troops. It can have assigned and deassigned workers. Higher production accelerates the creation of troops. Average cost of construction. Average HP and general defence. If demolished, all workers become unemployed.

Road

Can not be damaged. Can be demolished.

Optional feature

Allows faster movement between connected road hexagons. Low cost of construction.

Wooden walls

Average cost of construction. High HP and general defence. Can be demolished.

Wooden towers

Shoot nearby enemies in the area. High cost of construction. High HP and general defence. Can be demolished. It has the additional properties of damage, range and attack frequency. Attacks automatically nearby enemies.

Stone walls

High cost of construction. Very high HP and general defence. Can be demolished.

Stone towers

Shoot nearby enemies in the area. Very high cost of construction. Very high HP and general defence. Can be demolished. It has the additional properties of damage, range and attack frequency. Attacks automatically nearby enemies.

Spawn building

Non constructible building. Spawn troops nearby if there is space nearby every space of time.

Lord's castle

It is a subclass of spawn building. Objective to be destroyed in every level.

3.2.2.3.3 Healing

Buildings heal over time until they reach their maximum Hit-points.

3.2.2.4 Combat

When two troops fight, the attacking troop uses its charge attack in its first attack instead of its normal attack, if it has a charge attack, and uses the charge attack animation. Units of the troops will make an attack animation in an interval defined by its attack frequency and deal damage to the enemy troop. Before the attack, the outcome HP of the attacked troop is calculated and it is divided between the max units of the troop. If the outcome plus one is lower than the actual unit size, a unit or more will be killed during the animation attack, until the outcome is higher than the unit size.

Each troop is assigned to a determined hexagon, and if the troop is killed, its hexagon becomes empty.

If a troop is attacked during movement, the troop gets attached to the hexagon in which it is placed right now for some time.

3.2.3 Screenflow

3.2.3.1 Screen Flow Chart

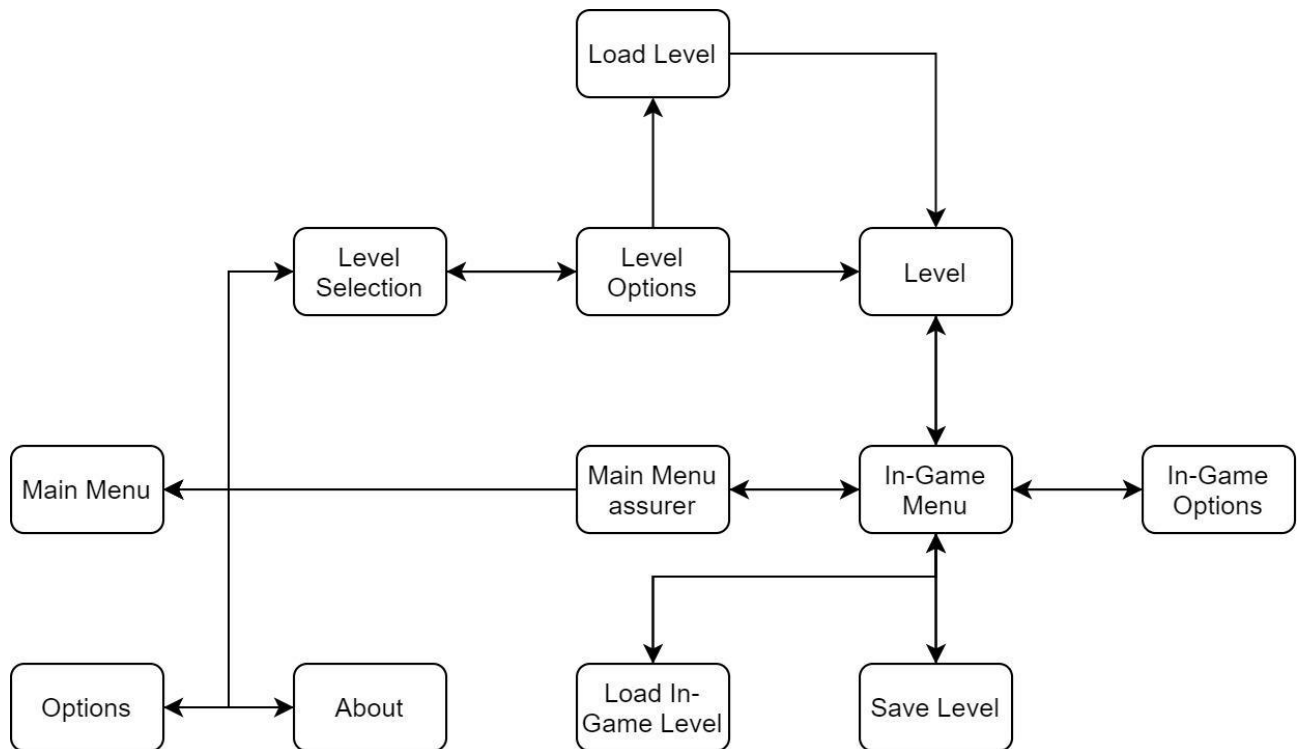


Figure 3.2: Screen Flow Chart

3.2.3.2 Screen Descriptions

Main Menu

Shows game title and is the start point when the game app is initialised. It also shows links to authors' web-pages.

Options

Allows general game configuration.

About

Shows additional information about the game.

Level Selection

Allows to select a specific level.

Level Options (Optional Feature)

Allows configuration of some features of the level.

Load Level (Optional Feature)

Allows to load a saved level.

Level

Environment where the gameplay is developed.

In-Game Menu

Gate to other menus during gameplay. It is accessed by the pause button in the in-game interface. Pauses the game.

In-Game Options

Allows general game configuration.

Save Level (Optional Feature)

Allows to save the progress of a level. The number of games that can be stored is limited.

In-Game Load Level (Optional Feature)

Allows to load a concrete game setted in the same level.

Main Menu Assurer

Asks again the player if he/she wants to go to the main menu.

3.2.4 Game Options

Allows to change the graphical quality of the game and to configure the sound.

3.3 Story, Setting and Character

3.3.1 Story and Narrative

3.3.1.1 Back story

The country is submerged in constant war. Lords are only interested in gaining power and the people suffer because of it. Rebellions spread all over the country and follow new leaders for promises of food, peace and freedom once they gain the power. And with this will and spirit, the cycle repeats once again.

3.3.1.2 Plot Elements

The Winter's uprising

The player leads the revolt against the local lord.

The Springs revelations

The power of the players grows, and attempts to face and fight the higher novelty.

The rising Summer

The player is ready to face the baron of his territory. The flame that his side has been carrying on has started corrupting and turning greener.

The Autumn decay

Player has ascended to power and is a rival to the rest barons. His flame has been completely corrupted, and now has to fight the other barons and the upcoming revolts.

3.3.2 Game World

3.3.2.1 General look and feel of world

The main religion of the country is similar to the Christian.

The country turns around the flame of god. It is a symbol of the connexion with God, consummation, purity, the holy spirit and faith between many others, and it lives in the heart of every spirit.

But the interpretation of the word of god is not unique, and there exists a strong separation between the interpretation of the elites and the local people. While the elites search for homogenization and a closer approach to the other countries, claiming as heresy the

approaches distant from their beliefs, the local people believe in an integration of their local culture and the faith in god.

This makes a distinguishable separation between the areas influenced by the nobility and the areas separated from it.

The areas influenced by the nobility have a high inspiration in medieval England while at the same time seems corrupted and damaged by the constant war. The flame that they pray has become green and their lords look like undead and act with tyranny. Many of their buildings are made of stone, but, specially the houses of their citizens, are made of wood and have roof tiles. The towns, outposts and cities tend to build around a big dark green crystal that resonates with necrotic energy.

The areas separated from the nobility are more inspired in the Viking culture. They live in wooden and more modest buildings and they more often, but not always, pray to a more natural, warmer and more destructive flame.

3.3.2.2 Factions

3.3.2.2.1 Lords of the ashes

Faction of the AI.

Back story

High nobility and clergy of the country. Their bodies are as corrupted as their souls. They use the power of the corrupted flame and the undead to control their people.

Personality

Arrogant and ambitious. They see human life as just a resource.

Look

Their skin is pale, and their expression shows a lack of empathy, anger, desperation and/or hunger. Green, dark and grey are their main colours. They are heavily inspired by the medieval English nobility and clergy.

Troops animations

Troops of this faction tend to have aggressive, strong and thoughtless movements.

3.3.2.2.2 Followers of Moses

Faction of the player

Back story

Population and low nobility and clergy that are situated away from the centres of power that decided to rebel against the lords of the ashes. Their rebellion is resembled by themselves with the rebellion of Moses against the pharaohs. Some of them wear horns on their helmets, not only as a symbol of their culture, but also as a symbol of the moment where Moses was given its revelations, as in some translations of the bible, Moses gained horns as a symbol of his received knowledge when God talked to him. They are heavily inspired by the popular perspective of the Vikings culture.

Personality

Passionate, aggressive, fervent and thoughtless.

Look

Barbarian, close to tribal and Viking. They are symbolized with red, orange and close warm colours.

Troops animations

Troops of this faction tend to have aggressive and frightened movements, as they fight for the cause but also for their life.

3.4 Levels

3.4.1 Level #1

3.4.1.1 Synopsis

The rebellion has begun. Rumours are spreading around to the nearby village and the local lord is going to react soon. The player has to recollect resources and destroy the lord's castle in order to stop the attacks.

The level presents the world to the player and its mechanics.

3.4.1.2 Introductory Material

Resources

Wood, settlers, food and iron.

Troops

Thrasher, hunter, swordsman, zombie and knight.

Buildings

Settlers centre, pot, sawmill, hunting shack, iron mines, shack, forge, road, wooden wall, wooden tower, stone wall, stone tower, spawn building and lord's castle.

3.4.1.3 Objectives

The main objective is to destroy the enemy's main base and progress through the story.

The side objectives are constructing buildings, collecting resources, creating troops, exploring and destroying enemy troops, buildings and outposts.

3.4.1.4 Physical Description

The level is set in winter.

There are 3 main areas in the map:

Coastline

Area where the player starts. Part of the coastline is frozen, but the water is still passable for ships. It doesn't occupy much space, the player will have to move to the forest.

Forest

It is the biggest area of the map. Full of trees, most of them dead, and also with a relevant relief. There are walkable and buildable places, but there are limited and form paths of different sizes inside the forest. Iron and outposts can be found inside.

Hills

Rugged and mountainous terrain, it is the place where the lord's castle is found. It has stone walls, towers and buildings, but most of them are devastated.

3.5 Interface

3.5.1 Visual System

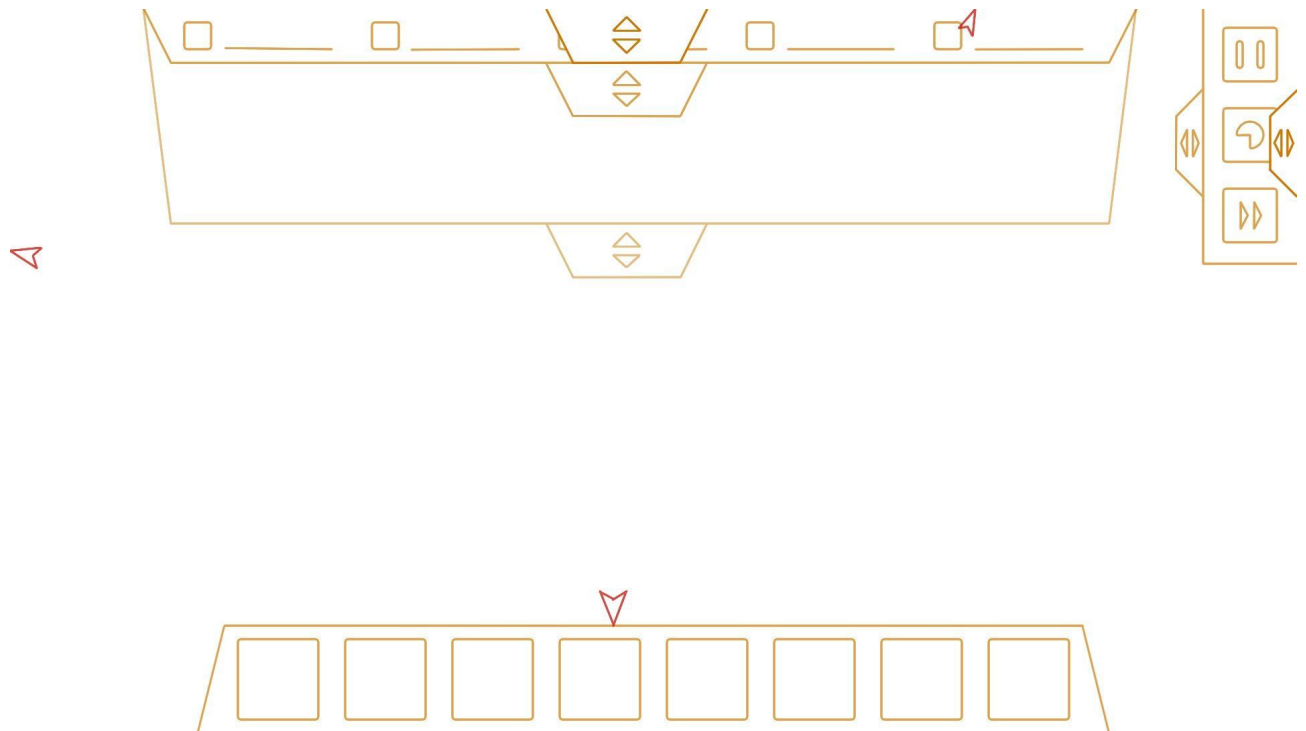


Figure 3.3: Visual representation of the user interface

The player uses an aerial view camera.

The game-world is separated with hexagons, each with its terrain type and a possible resource or troop or building that, if present, can be selected.

The game view is oriented from a horizontal perspective, and has a three level interface as shown in the figure 3, ordered from darker to clearer. First level only shows the sliders that can be dragged and dropped to show its next level of interface or to go back to the first level. In the second level, the middle-up interface shows basic information about the selected troop, the resources needed to build the selected building and the present resources of the player or the present resources of the player only. This interface can be dragged more to show additional information if it is present.

The top-right interface shows the pause button, the button that slows down the time and the button that speeds it up, when it is dragged.

The middle-down interface only shows when a hexagon is selected, and it shows the list of buildings or, if there is a building in the hexagon, the different actions that this building can elaborate. If there are not enough resources for the action, the icon of the needed resources will show a red border and throb. Additional functionalities can be added.

Optional feature

The red arrows indicate conflict areas, where a troop or building is being attacked.

3.5.2 Control System

The player can move through the map using his finger, by holding down the finger and moving it to the opposite direction he/she wants to go, or the cursor, by moving the cursor to the border of the screen limits in the desired direction.

A hexagon or element on a hexagon is selected when the player taps or clicks on the hexagon. Elements on the middle-down interface can also be selected. This will show the needed resources to realize the action near the actual resources. If there are enough resources, if the element is clicked or tapped again the action will be executed.

To deselect a hexagon or an element inside a hexagon, this has to be tapped or clicked again or the player has to select another hexagon.

If a troop is selected and a different hexagon than the one that the troop is located is tapped or clicked, the troop will move to the selected hexagon and the troop is deselected.

If a troop is selected, a checkbox appears on the middle-top interface. This is the “advance and attack” checkbox and can be tapped or clicked to change the attribute and by this, the behavior of the troop.

Optional feature

Clicking on red arrows moves the camera to the area indicated.

3.5.3 Audio

3.5.3.1 Music

The soundtrack is heavenly inspired by pagan music, mixed with classical and electronic music. Because of this, the “Warhammer Vermintide” saga soundtracks are a big reference. It separates into 4 moods:

Peace

This music plays mainly while the player is constructing buildings or collecting resources. The main emotion of this music is peace, but can have some elements of tension and terror to remind its fragility to the player as he/she is facing a stronger and more terrifying force. It can be replaced by ambient sounds.

Exploration

This music plays while the player is exploring. It is still peaceful, but the element of tension and terror is much more present.

Tension

This music plays while an attack is approaching or there is a small conflict (4-8 troops fighting). It is remarkably tense and has touches of brutality, psychedelic and terror.

Great combat

This music plays during big combats or the attack on the lord’s castle. It has similar emotions to the music for tension, but it is much more intense and has touches of epic music. It sounds more of a desperate fight than an epic fight.

The music has a smooth change when the mood is changed.

3.5.4 Sound Effects

Forest ambient, sea ambient, horror ambient, fantasy UI, medieval combat, spells, construction, wood work, mine work, hunt work, footsteps, human body, stone and (optional) troop identifiers lines.

3.6 Artificial Intelligence

3.6.1 Opponent AI

The opponent IA is responsible for determining the size of the attacks, the regularity of the attacks and which main path will the attackers or horde take.

3.6.2 Support AI

If the troop is set “advance and attack”, option activated by default, in the process of moving, if the troop sees any enemy in their vision area, it will move to attack the closest available enemy. If all the enemies available to attack are too far, the troop will go to the closest hexagon to the enemy available and once it has reached the position, it will set its status to waiting and check again if it is able to attack any enemy. If not, it will wait until a troop dies and then check again if there is any way to attack an enemy. If the enemy dies, the troops will choose another enemy or, if it has a hexagon to move assigned, it will move to the hexagon.

3.6.2.1 Player and Collision Detection

Every troop has a collider the size of a hexagon.

3.6.2.2 Pathfinding

The troop uses the pathfinding implemented in unity, but it also helps itself with a matrix of the map, as all of the positions that the pathfinding receives come from this matrix. This matrix also stores information about the terrain. Mainly, this information is the type, troop, resource, building that is placed in this hexagon or if it is a non-walkable hexagon. The pathfinding can be replaced by an own A* algorithm that moves exclusively in the directions of the normals of the hexagons.

If the troop is moving, it doesn't block the movement of other troops until it stops moving.

4 System Analysis and Design

This chapter presents the requirements analysis, design and architecture of the proposed work, as well as, where appropriate, its interface design.

4.1 Requirement Analysis

As this is a game planned to be playable on both PC and mobile devices, the input system has to support controls from bough devices. Also, the performance of the project should not be too high in order to be used on mobile devices.

The game is set in a dark fantasy world inspired by the Viking, the English medieval culture and “Warhammer Fantasy”, specifically from the baroque and overloaded static from the last one. At the same time, the visual style is a low poly one, specifically the one used by “Synty Store” in its “polygon” style.

Because the project is a prototype and not a fully developed game, screens that do not present directly core elements of the project are not necessary, leading to the first level as the only required scene. In this scene, a hexagonal grid that extends through all the reachable terrain is displayed. In this hexagonal grid, the player is able to construct buildings by using the user’s resources and selecting a cell from the grid. A cell from the hexagonal grid is selected by clicking on PC and by tapping on mobile devices at the desired hexagon. To build something in the cell it must be empty, otherwise, a troop or a building will be selected and a UI element that indicates the attributes of the selected entity will pop up. Depending on the building, an action panel from the building may also pop up.

Buildings allow the player to produce more resources, defend the base and create troops. The player can construct buildings by selecting an empty cell of the hex grid and interacting with the buttons of the panel that pops up after selecting the cell. These buttons are special, not only because each of them represents a building, but also because they are “Double interaction buttons” as some actions need a button to be pressed twice in order to improve the control of the player on the game and to perform different actions depending on the interaction. This happens also when creating troops. In the first interaction, the resources needed to perform an action and the information relevant to the action appear on the screen. In the second interaction in a row with the same button, the entity is created. An interaction occurs by clicking on PC and by tapping on mobile devices on the button.

A building that stands out from the rest is the “Settlers Center” as the player needs to have at least one in order to not lose the game. Also, these buildings are very important as they can create a very important element of the scene, the troops, which are able to destroy the troops and buildings of the enemy faction including the “Lord’s castle”, a building that needs to be destroyed to win the game.

The player can move the troops by selecting a hexagon while having a selected troop. This is made by clicking on PC and by tapping on mobile devices on an empty hexagon. Also, the player can order to attack a troop or unit from another faction. To do it, troops will move to attack an enemy troop if they are not in range to perform an attack. When they are close enough, they will start combat with the other troop if they are in a nearby hexagon or the attacking troop will perform a ranged attack. This is made by clicking on PC and by tapping on mobile devices on a hexagon with an enemy troop or building in it.

Troops also will attack any nearby enemies if they are set to have this behavior or are able to do it. This behavior is shared with buildings.

The enemy faction has special behavior for its troops, in which these troops will try to reach a target area attacking all the enemies in the way (in development).

It is crucial to point out that the player is also able to move the camera by dragging it. This is performed by clicking on PC or touching in mobile devices and moving the cursor in the opposite direction.

4.1.1 Functional Requirements

Taking into account the previous analysis, the following functional requirements can be defined:

R1: The player can select cells from the hex grid

R2: The player can select troops

R3: The player can move selected troops

R4: The player can attack enemy troops and buildings

R5: The player can interact with buttons that have two different feedbacks depending on how many times in a row the button has been pressed

R6: The player can generate troops

R7: The player can win the game

R8: The player can lose the game

R9: The player can generate resources

R10: The UI changes depending on the selected entity

R11: There are different UI elements for all the different types of entities and actions

R12: The player can create buildings

R13: The player accumulates resources

R14: The player can use resources

R15: The player can generate resources

R16: The player can move the camera

R17: The player can select buildings

4.1.2 Non-functional Requirements

The qualities that the project must fulfill are:

R18: The game esthetics baroque and dark

R19: The game is able to be played on PC and mobile devices

R20: The project does not require high performance

R21: The game sounds and soundtrack feels the atmosphere of the game

R22: The controls are easy and intuitive

R23: The attributes from entities are easy to understand

4.2 System Design

In this part the design of the system is explained by cases of use, based on the functional requirements, and the case use diagram.

Requirement fulfilled	R1
Actor	Player
Description	The player can select cells from the hex grid
Preconditions	None
Normal Sequence	PC: <ul style="list-style-type: none"> • The player clicks on a cell of the grid Mobile: <ul style="list-style-type: none"> • The player taps on a cell of the grid
Alternative Sequence	None

Table 3.1: Case of use «CU01. Select cell from the hex grid»

Requirement fulfilled	R2
Actor	Player
Description	The player can select troops by selecting the hex grid cell in which the troop is located
Preconditions	The cell of the grid that the player is going to selected does have a troop on it
Normal Sequence	PC: <ul style="list-style-type: none"> • The player clicks on a cell of the grid Mobile: <ul style="list-style-type: none"> • The player taps on a cell of the grid
Alternative Sequence	None

Table 3.2: Case of use «CU02. Select troop»

Requirement fulfilled	R17
Actor	Player
Description	The player can select buildings by selecting the hex grid cell in which the building is located
Preconditions	The cell of the grid that the player is going to selected does have a building on it
Normal Sequence	PC: <ul style="list-style-type: none"> • The player clicks on a cell of the grid Mobile: <ul style="list-style-type: none"> • The player taps on a cell of the grid
Alternative Sequence	None

Table 3.3: Case of use «CU03. Select building»

Requirement fulfilled	R3
Actor	Player
Description	The player can move selected troops to an empty target hex grid cell.
Preconditions	The player has selected a troop from its faction
Normal Sequence	PC: <ul style="list-style-type: none"> • The player clicks on an empty cell of the grid Mobile: <ul style="list-style-type: none"> • The player taps on an empty cell of the grid
Alternative Sequence	None

Table 3.4: Case of use «CU04. Move troop»

Requirement fulfilled	R4
Actor	Player
Description	The player can attack enemy troops and buildings when selecting the hex grid cell where the enemy entity is located if the player has previously selected a troop.
Preconditions	The player has selected a troop from its faction
Normal Sequence	PC: <ul style="list-style-type: none"> • The player clicks on a cell of the grid that has an enemy entity on it Mobile: <ul style="list-style-type: none"> • The player taps on a cell of the grid that has an enemy entity on it
Alternative Sequence	None

Table 3.5: Case of use «CU05. Attack enemy»

Requirement fulfilled	R6
Actor	Player
Description	The player can generate troops using the “Settlers Center” building types by their pop-up panels.
Preconditions	The player has selected a “Settlers Center”
Normal Sequence	<p>PC:</p> <ul style="list-style-type: none"> ● The player clicks on one of the buttons from the middle down panel ● The player clicks again on this button <p>Mobile:</p> <ul style="list-style-type: none"> ● The player taps on one of the buttons from the middle down panel ● The player taps again in this button
Alternative Sequence	<p>PC:</p> <ul style="list-style-type: none"> ● The player clicks on one of the buttons from the middle down panel ● The player clicks on a different button from the last clicked on this panel. This action can be repeated indefinitely. ● The player clicks again on this button <p>Mobile:</p> <ul style="list-style-type: none"> ● The player taps on one of the buttons from the middle down panel ● The player taps on a different button from the last tapped on this panel. This action can be repeated indefinitely. ● The player taps again in this button

Table 3.6: Case of use «CU06. Create troops»

Requirement fulfilled	R12
Actor	Player
Description	The player can create buildings on empty cells using the construction selection panel
Preconditions	The player has selected an empty cell
Normal Sequence	<p>PC:</p> <ul style="list-style-type: none"> • The player clicks on one of the buttons from the middle down panel • The player clicks again on this button <p>Mobile:</p> <ul style="list-style-type: none"> • The player taps on one of the buttons from the middle down panel • The player taps again in this button
Alternative Sequence	<p>PC:</p> <ul style="list-style-type: none"> • The player clicks on one of the buttons from the middle down panel • The player clicks on a different button from the last clicked on this panel. This action can be repeated indefinitely. • The player clicks again on this button <p>Mobile:</p> <ul style="list-style-type: none"> • The player taps on one of the buttons from the middle down panel • The player taps on a different button from the last tapped on this panel. This action can be repeated indefinitely. • The player taps again in this button

Table 3.7: Case of use «CU01. Construct a building»

Requirement fulfilled	R16
Actor	Player
Description	The player can move the camera by dragging its cursor
Preconditions	None
Normal Sequence	PC: <ul style="list-style-type: none"> • The player holds the click button and use the cursor to drag the camera • When the player stops holding the click button the camera stops moving Mobile: <ul style="list-style-type: none"> • The player touches the screen and drag the camera • When the player stops touching the screen the camera stops moving
Alternative Sequence	None

Table 3.8: Case of use «CU08. Move the camera»

Requirement fulfilled	R7
Actor	Player
Description	The player can win the game after destroying the Lord's castle
Preconditions	A troop is attacking the Lord's castle
Normal Sequence	One of the troops that attack the Lord's castle survives until it has inflicted enough damage
Alternative Sequence	None

Table 3.9: Case of use «CU09. Win the game»

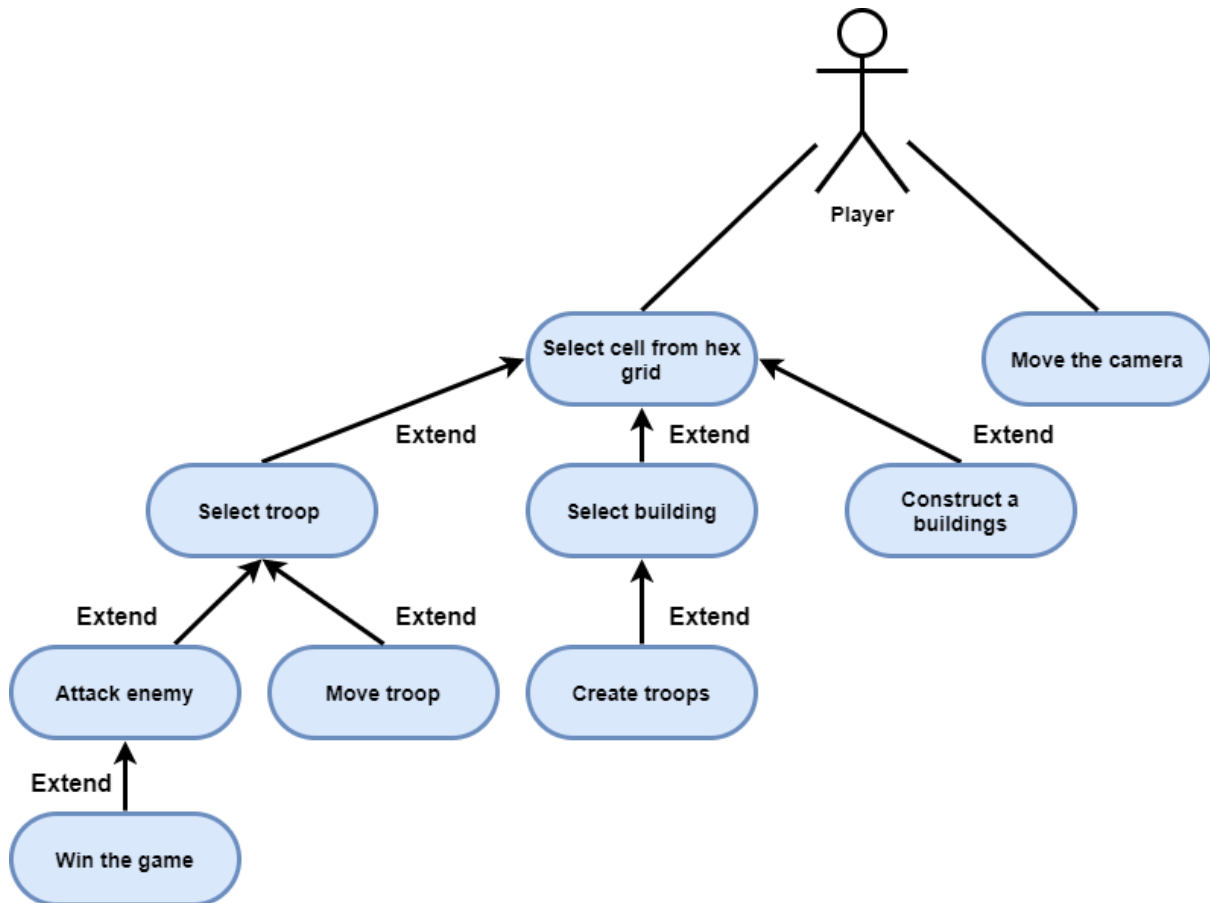


Figure 4.1: Case use diagram (made in draw.io [10])

4.3 System Architecture

According to the unity documentation, the system requirements depend on the device:

4.3.1 Android

As android is the only operating system that I can test that the application works properly, is the only operating system for which the prototype will be built on. The requirement for this device are;

- **Version:** 4.4 (API 19)+
- **CPU:** ARMv7 with Neon Support (32-bit) or ARM64
- **Graphics API:** OpenGL ES 2.0+, OpenGL ES 3.0+, Vulkan
- **Additional requirements:** 1GB+ RAM.

Supported hardware devices must meet or exceed Google's Android Compatibility Definition limited to the following Device Types:

1. Handheld (Section 2.2)
2. Television (Section 2.3)
3. Tablets (Section 2.6)

Hardware must natively be running Android OS. Android within a Container or Emulator is not supported.

4.3.2 Desktop

According to the unity's page, the application can be used in the devices with the following requirements;

- **Operating system version:** Windows 7 (SP1+), Windows 10, High Sierra 10.13+ and Ubuntu 20.04, Ubuntu 18.04, and CentOS 7.
- **CPU:** x86, x64 architecture with SSE2 instruction set support.
- **Graphics API:** DX10, DX11, DX12 capable on Windows, Metal capable Intel and AMD GPUs on Mac and OpenGL 3.2+, Vulkan capable on Linux.
- **Additional requirements:** Hardware vendor officially supported drivers on Windows, Apple officially supported drivers on apple and on Linux Gnome desktop environment running on top of X11 windowing system. Also, in Linux, other configuration and user environment as provided stock with the supported distribution (such as Kernel or Compositor), Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver.

4.4 Interface Design



Figure 4.2: UI of the Prototype

The player uses an aerial view camera.

The game world is separated with hexagons, each with a possible troop or building that, if present, can be selected.

The game view is oriented from a horizontal perspective and has a dynamic interface that changes the elements, and with this also the number of elements, depending on the selected object.

The middle-top interface gives related information to the available actions, which includes 7 different panels in total. Those are the panel about the selected troop, the panel of the troop that is going to be recruited, the panel of the building, the panel of the construction of the building, the panel of the current player's resources, the panel of the available buildings to be constructed and the panel of the available troops to be created.

There have been noticeable changes between the designed UI in the game design document (GDD) and the final UI. The draggers that allowed to change the amount of information on the screen have disappeared. They were an interesting element of the UI but not an essential part of it, and its utility needed to be tested.

The middle-down interface only shows when a hexagon is selected, and it shows the list of buildings or, if there is a building in the hexagon, the different actions that this building can elaborate. The player can scroll through the different elements of the lists.

5 Work Development and Results

The main tool for this project has been Unity [3], and most of the work has been related to it. When I have not found any way to progress or had trouble doing it, I have always drawn on Unity's manual [7], Unity's forum [4], Stack Overflow [1] forum and Youtube [6] video game tutorials and devlogs. Also, the fundamentals of the Hex Grid UI are based on the "Catlike Coding" "Hex Map" tutorials [5] the code of these elements derives from it.

I also used Google docs [2] to take notes and write documents for the project and checked its correct writing with Grammarly [11].

5.1 Work Development

5.1.1 Game design document

The development of the project started with the game design document (GDD). The core elements of the game have to be interesting and the game has to have a cohesive and immersive world-building, even if the game is using assets, in order to build a useful prototype and concept, so I took my time on it.

This document was written on Libreoffice [8].

5.1.2 Setting up the project and the Github problem

After finishing my GDD, one of the main problems appeared when I was setting up the project and implementing the Github repository. In short, the official asset from Github was corrupting my project and I spent a week searching for the problem until I discovered it. Finally, I decided to just use GitHub desktop.

Also, I imported the Synty Store POLYGON Knights Pack [12], Adventure Pack [13], Particle FX Pack [15] and Vikings Pack [14], and installed the Peek editor toolkit [17] to help me work more efficiently.

5.1.3 Hex grid

When I solved my problems with Github, I started with the research of information on how to create a hexagonal grid. After finding different options and reading different tutorials I decided to use the first chapter of the tutorial “Hex Map” from the “Catlike Coding” page [5] and progress with the project. In order to use the “Hex Map” script on the project, I had to first learn how the universal render pipeline (URP) and the new unity input system works and adapt the source code to it. I also added the Hex grid projection that the author describes in chapter 15 of his tutorial.

The source code of the “Hex Map” was designed to evolve into, as the name indicates, a map that is generated randomly with different types of terrains and is directly the place where the game is played, not only an interface. Because of this, I had to modify the creation script of the source code so it generated all of its vectors above a certain distance from a given terrain. The result of the original script and my described modifications can be seen in figures 4.3.1 and 4.3.2.

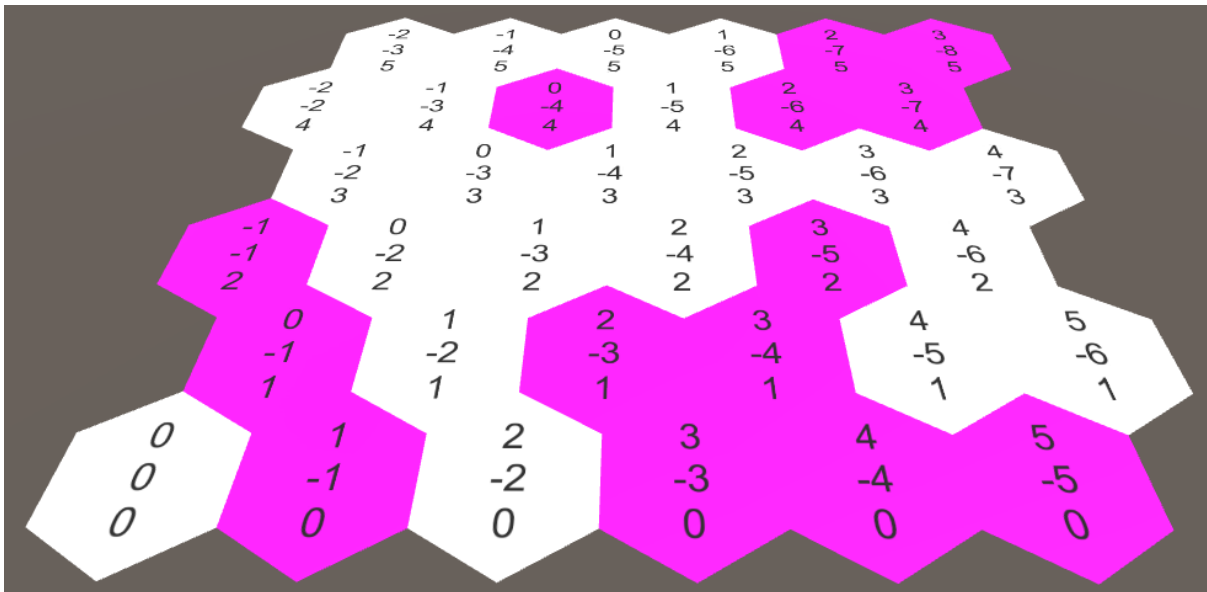


Figure 5.1: Original “Hex Map” Script Behaviour

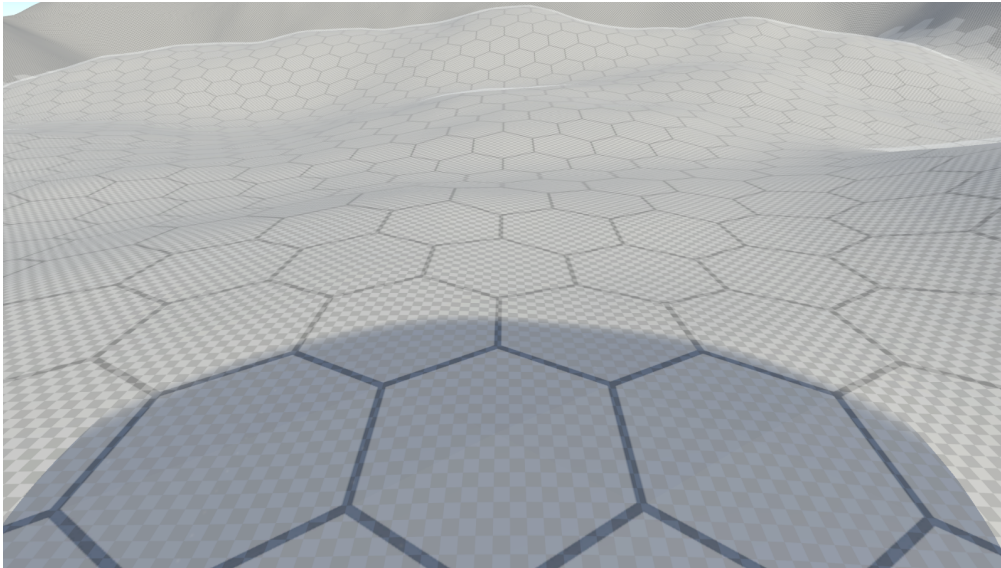


Figure 5.2: My modification to “Hex Map” Script Behaviour

5.1.4 Special terrain types and resources

As there were more relevant tasks of the project than implementing a system with special terrains and hex grid cells with resources, these elements had not been implemented. The buildings that depended on this system do not depend on it in the prototype.

5.1.5 Troops and units

The next objective to accomplish was to develop the troops of the game and its behavior. The script implementation of the troops, and the units that compose them, took much more than expected due to its complexity.

5.1.5.1 Base development

First, I selected an appropriate model from the Synty Store assets and searched for animations in Mixamo [9]. Following that, I created an animation controller with a structure that could be adapted for all troops and added the searched animations to this controller.

All the units are controlled and grouped by troop game objects, which move them to designated places by code or by the selected cells of the hex grid.

With these troop game objects it is archived that the soldiers of the project move in formations and perform actions in groups, having a similar squad behavior of the troop's groups from the “The Settlers V: Heritage of Kings” game.



Figure 5.3: Troops groups from “The Settlers V: Heritage of Kings” game



Figure 5.4: Black Hole Sun troops in formation

5.1.5.2 Troop attributes

All troops when are created take information from an object, specifically a “Scriptable object”, that stores the different models a unit from the troop can have, the formation on the units, that includes the information to position the units independently of the number of units that the troop beholds, the projectiles that the unity launches, if needed, and the attributes of the troop with are: hit points, units, range of attack, range of damage inflicted per attack, range

of defense, attack frequency, faction, food maintenance and advance and attack status. Each troop stores the cost of its creation. The food maintenance has not been implemented as there were more relevant features to be completed.

5.1.5.3 Troops behavior

The behavior of the troop includes:

- **Attacking a target**

This can drift to the actions of attacking a building or attacking a troop, which in both cases implies moving to the target until it is in range for an attack and then giving targets to the troop's units and, unless the troop is not on melee range, sending the units to move close to the enemy units, when attacking a troop, or the target, when attacking a building. After this, each unit is ordered to perform attacks to its target.

Once the troops are attacking, it inflicts damage to the enemy troops based on its attack and its attack frequency. This means that the combat between two entities is essentially an illusion, similar to what happens in the Total war: Warhammer [25] game.



Figure 5.5: Total war Warhammer [26] troops (Screenshot taken from Steam [27] store)

- **Taking damage**

A troop can receive damage that will drop its HP from any source. Depending on the damage received and the current HP, it may imply that a unit is marked as 'ready to die', which means that this unit will die in the next attack target to it. Fewer units also imply less damage inflicted and less defense.

- ***Managing the destruction of the objective***

When the objective of the troops is destroyed, the troop sends all of its units to the formation and then repositionates the troops in the nearest hexagonal cell, unless the troops have been attacked during combat with a melee assault. In this case, the troop changes its objective to one of the attackers.

When the objective of a unit is destroyed, the troops search for a new objective for the troop in the enemy unit's lists so it can continue fighting.

- ***Assignating the objectives to units***

When the troop is performing an attack to another troop, every unit from the assaulting troop has a unit from the enemy troop assigned as a target. If there are more attacker units than the ones that are defending, multiple units may have the same target.

- ***Managing the movement of the troop***

It includes following a target, reaching a position and reacting when the troop has been attacked during displacement. The pathfinding of the game uses Unity's [3] pathfinding solution that utilizes navigation agents, navigation maps and navigation obstacles to be avoided.

- ***Giving troop information***

The troop can provide its attributes in a text format when requested. It is especially used by the UI.

- ***Managing, partially, the animation of its units***

The troop script manages the animations that the units are using when the troop objective or the unit objective is destroyed, but the rest of the time the animations of the unit are decided by themselves.

- ***Advancing and attacking***

If the troop has the "Advance and attack" attribute set, it will decide to move and attack any nearby enemy by itself. Because other tasks had a higher priority the player is not allowed to decide if the unit has this attribute set using the user interface.

5.1.5.4 Units behavior

The behavior of the unit includes:

- ***Preparing for combat***

When units are going to assault an enemy unit, the defending unit waits for the attackers and activates itself as an obstacle for the navigation agents. The objectives of the attacking units and defenders are also set. After this, if the attackers are going to perform a melee assault, the attacking units are sent to come close to their target. Finally, the attacker starts performing attacks. If the defender has an enemy in melee range it also starts performing melee attacks.

When the units are going to assault a building they have similar behavior, however in this case the defending entity does not change its behavior and all the troops get ready to attack only one entity, the building.

- ***Performing attacks***

When performing attacks, the animation controller of the units is set in a combat status, where it faces its enemy and plays a set of animations waiting until it is time to perform an attack, that implies an attack animation, which is decided by the attack frequency of the troop plus a random amount of time lower than a second. When the unit performs the attack, if the enemy unit is set to die in the next attack, the enemy unit will be destroyed after the attack is launched, and the unit will call the troop for new instructions.

- ***Managing its movement***

During combat, each unit calculates its destinations set by the troop script and uses its navigation agent to move to those places. Also, the troop manages that, when it is not moving, it has set as a navigation obstacle for the rest navigation agents.

- ***Managing its enemies***

Each unit has a list of enemies that are currently fighting and it is responsible for removing the destroyed enemy units.

- ***Managing its animations***

The unit is in charge of managing that when the unit is moving, a movement animation is performed according to the unit velocity and when it is not moving an idle animation is performed. Also, it has to manage when the unit enters combat, changing the unit between the attack state and the defending state. In the first state, it selects a random attack animation to be performed. In the second state, it chooses a random defense or idle animation to be performed.

- ***Launching projectiles***

If the troop of the unit has projectiles attached, the unit will throw projectiles when attacking. The launching of a projectile is performed by applying a velocity vector to the game object that is calculated by a parabolic equation that depends on the position from where the projectile is going to be launched and the objective position. The same method is applied when launching projectiles from buildings.

- ***Updating its position according to the terrain***

The unit sets the position of its height based on the height of the terrain on which it is every frame.

The behavior of the troops and units may appear simple, but it does have a lot of implications. The objective of a troop or a unit may be destroyed while the troop or unit is on the way to it or while it is attacking it by another troop, unit or building. Also, a troop can be attacked while performing an assault to another troop. All of these happen in a context where troops can be attacked from melee and ranged attacks. These scenarios provoke that all the code must support that its target may disappear at any moment in any process, especially on coroutines, causing that every minor failure in recognizing that the target is not non-existent will provoke all the behavior of the troop to be blocked.

5.1.5.5 Types of troops

Thrasher



Figure 5.6: Thrasher troop design

Basic troop. Low-average attack damage, low defense, high HP, high total unit maximum, melee range, average movement speed, low food maintenance and average attack frequency. Needs wood, food and settlers to be created.

Hunter

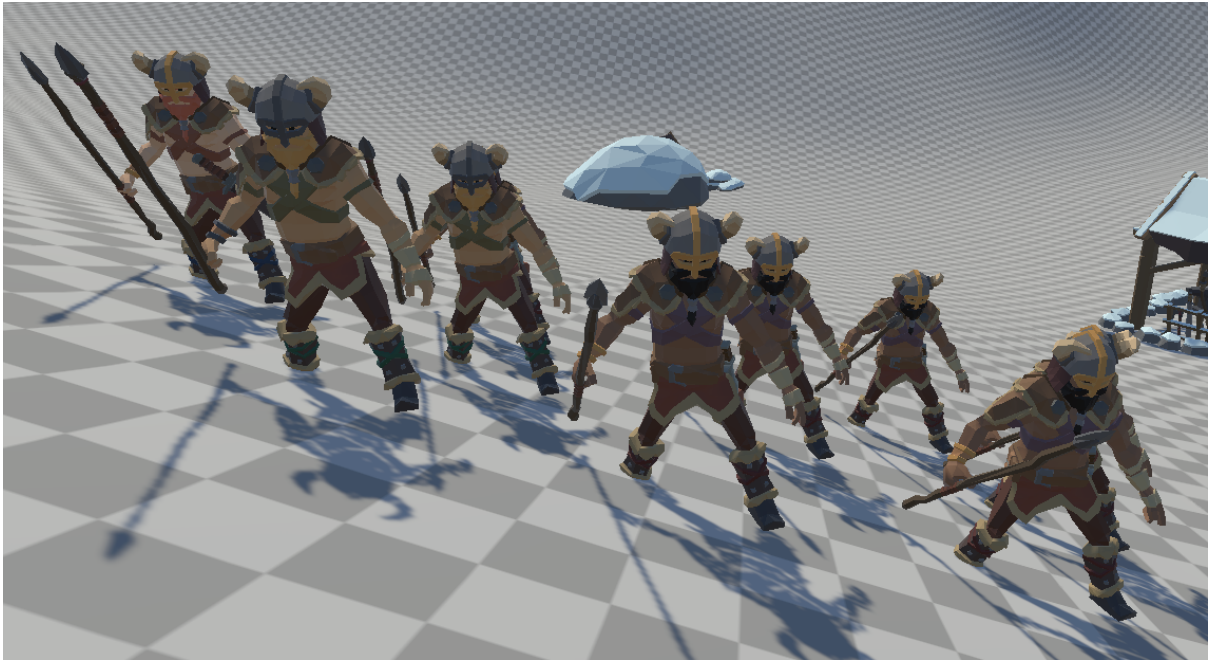


Figure 5.7: Hunter troop design

Basic troop. Low-average attack damage, very low defense, medium HP, average total unit maximum, range attacks, average movement speed, low food maintenance and low-average attack frequency. Needs wood, food and settlers to be created.

Swordsman



Figure 5.8: Swordsman troop design

Average troop. Average attack damage, medium defense, medium HP, medium total unit maximum, melee range, medium movement speed, average food maintain, no gold maintain and average attack frequency. Needs iron, food, wood and settlers to be created. Due to lack of time, it does not need a forge to be produced.

Zombie



Figure 5.9: Zombie troop design

Exclusive IA basic troop. Low attack damage, low defense, very high HP, very high total unit maximum, melee range, slow-medium movement speed, no food maintain, no gold maintain and average attack frequency.

Knight

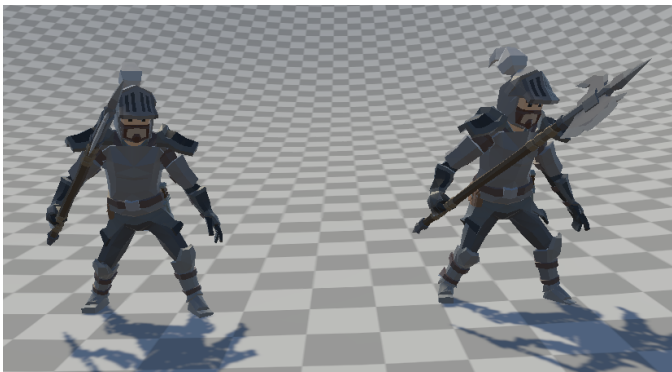


Figure 5.10: Knight troop design

Exclusive IA advanced troop. High attack damage, very high defense, average HP, low total unit maximum, melee range, slow-medium movement speed, no food maintenance, no gold maintenance and slow attack frequency.

5.1.6 Buildings

During the development of troops and other tasks, I have been creating the appearance of the buildings of the game using the Synty Store modular assets and, when the troop's and unit's scripts were ready for debugging, I started to develop the building script. It does have similarities with the troop behavior, so I decided that they would inherit from a common class, general entity. This class serves mainly as an interface of behaviors that both scripts share.

5.1.6.1 Attributes

Unlike troops, buildings store directly their attributes and any other necessary element. These attributes must include its hit points, its production, its defense range and its faction.

There are other main attributes, but these are optional: the range of its attacks, its attack range, its attack frequency and its projectiles.

5.1.6.2 Buildings behavior

Buildings are able to take damage and give information about their status, mainly for UI. Also, if the building has the required attributes to launch attacks, it will perform these and also manage which enemies enter and exit from its range. The projectiles from the will not mage damage until they have impacted.

Some of the features explained in the GDD are not implemented on the final prototype. This is the case of the workers' management of the buildings. Originally, the game was going to have a micromanaging system that allowed players to decide how many settlers work on a building. This is interesting when the player has a limited amount of settlers and also a limited amount of resources, making the user decide where the player is going to expand next. However, the prototype does not have limited resources, something that makes this mechanic pointless. Because of these, now the settlers are just a resource needed to construct these buildings.

Also, buildings do not heal over time, as there were more relevant tasks to be performed and I was not able to implement this feature properly.

The demolish functionality has not been implemented as there were more relevant tasks of the project that needed attention.

In order to fulfill the game's necessities, there have been different types of buildings developed.

5.1.6.3 Types of buildings

Settlers Center

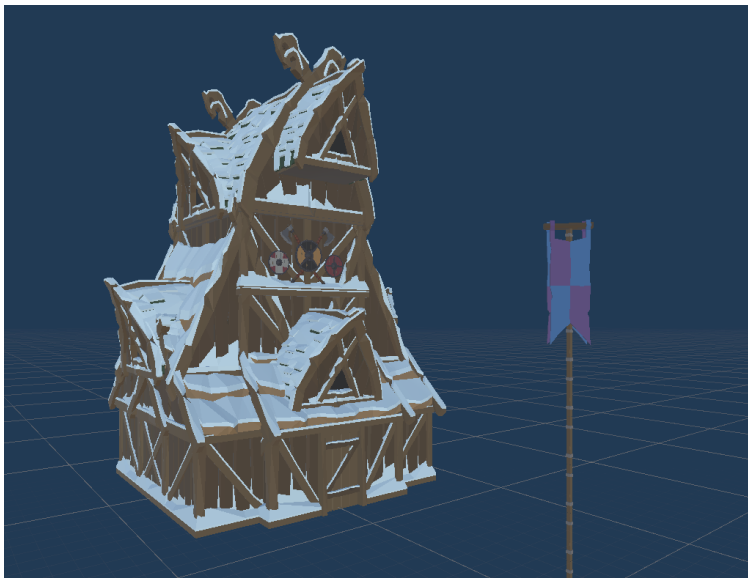


Figure 5.11: Settlers Center building design

Collects settlers and a small amount of wood. If all settlers centers are destroyed the player loses. It can create units on the position marked by a flag and can not be demolished. The position of the flag cannot be moved and the player cannot set a destination of where will the troop go when it is spawned, I decided this as there were other parts of the project that deserve more attention. High cost of construction. High HP and general defense.

The cost of building a Settlers Center does not elevate for every settlers center in the player faction as other tasks had more importance than this feature.

Port

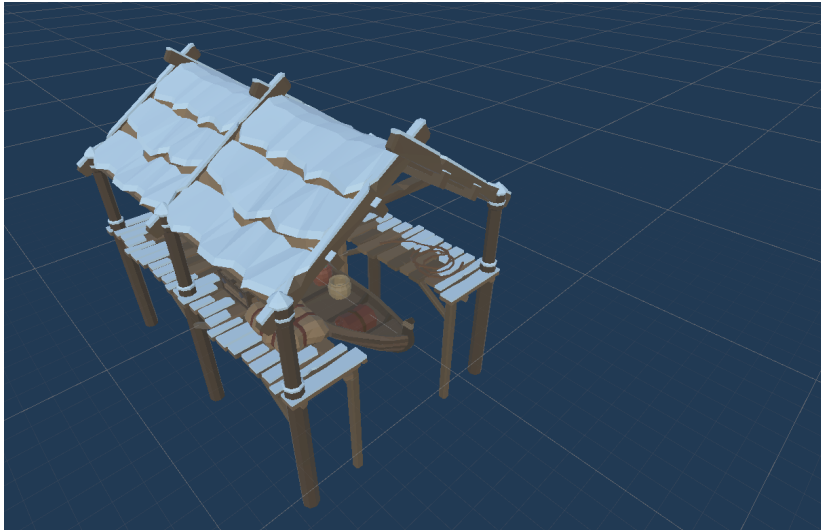


Figure 5.12: Port building design

Collects settlers and produces food and gold. High cost of construction. Average HP and weak general defense. Needs to be built on a water hexagon.

As these buildings have a lot of specific requirements, a level with the sea, a shader for the sea and a system to allow to identify when a hexagon is on a coastline, it is not worth the work to be implemented in the prototype, as it demands a lot of time that could be used in polishing the game.

Sawmill

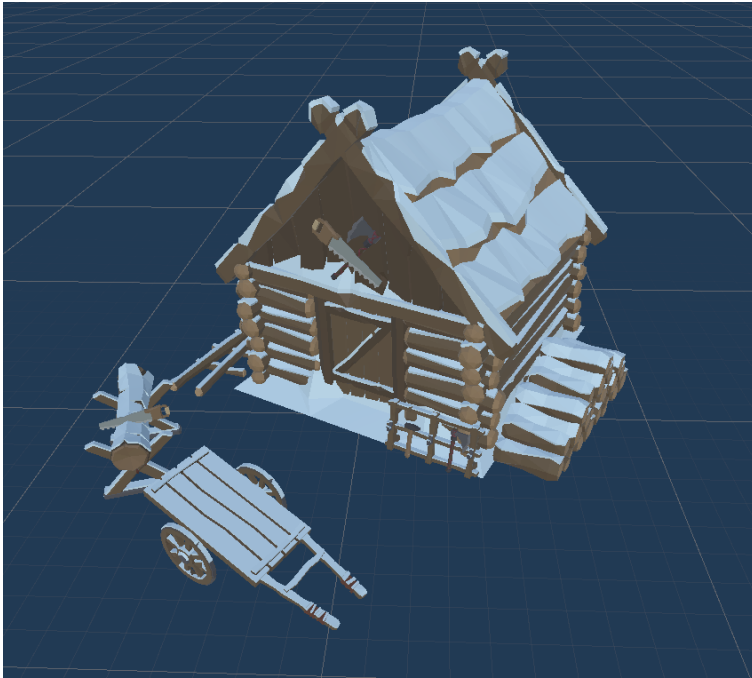


Figure 5.13: Sawmill building design

Produces wood. Low cost of construction. Average HP and general defense.

Hunting shack

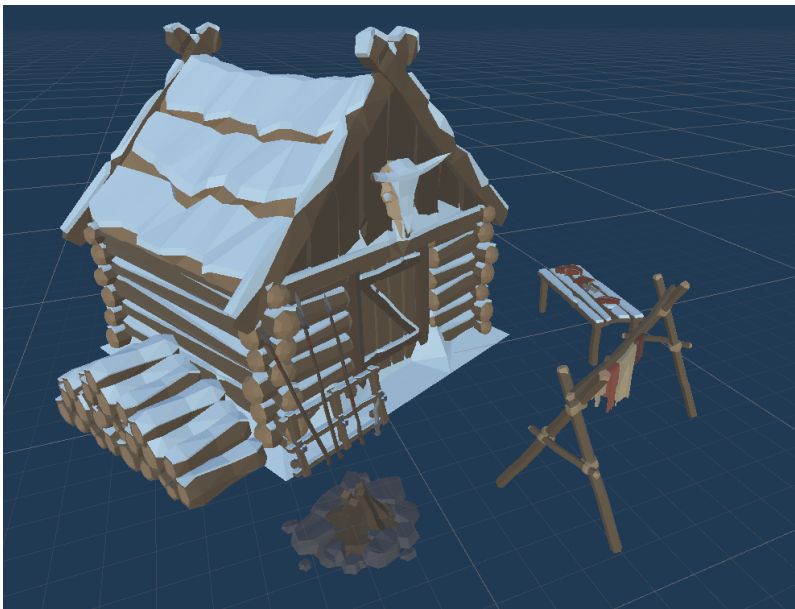


Figure 5.14: Hunting shack building design

Produces food. Average cost of construction. Average HP and general defense.

Iron mines

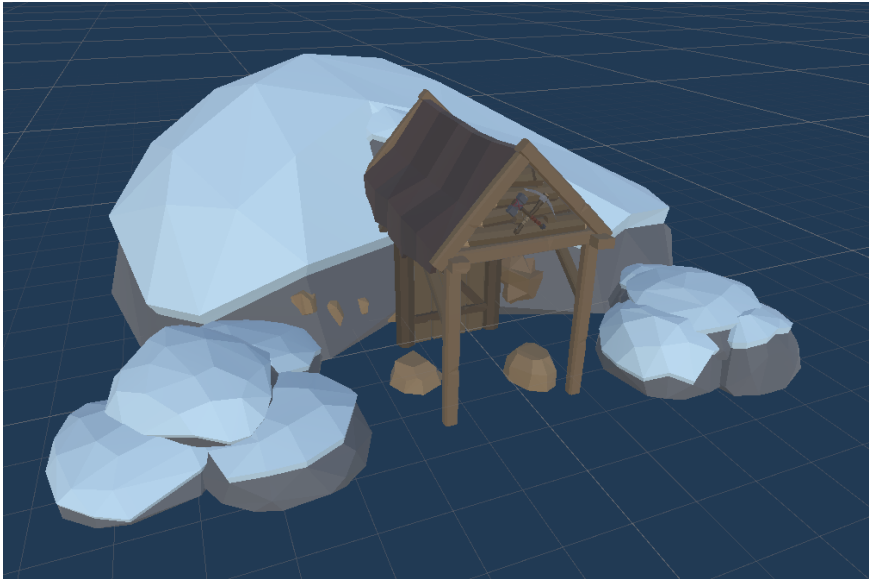


Figure 5.15: Iron mines building design

Collects iron. Need iron nearby. Average cost of construction. Average HP and general defense.

Forge

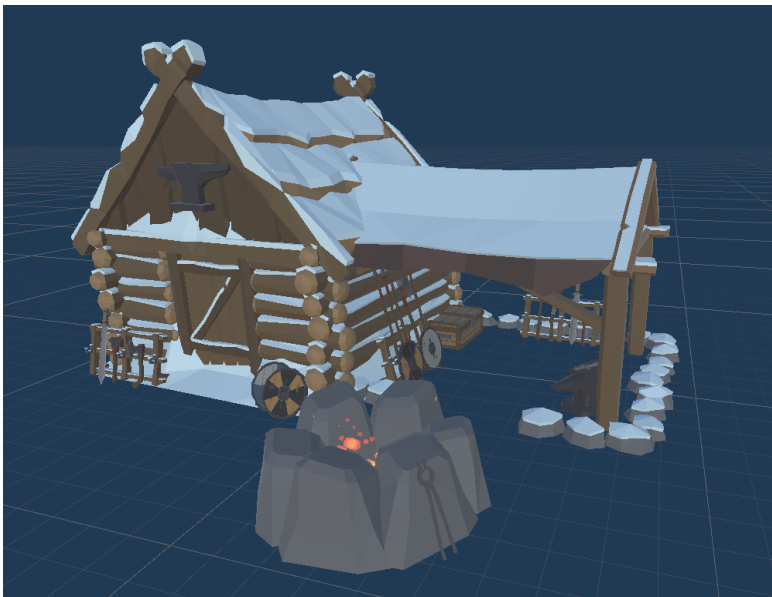


Figure 5.16: Forge building design

Needed for the production of specific troops. Average cost of construction. Average HP and general defense.

Road

Can not be damaged. Has mainly a decoration purpose.

Wooden walls

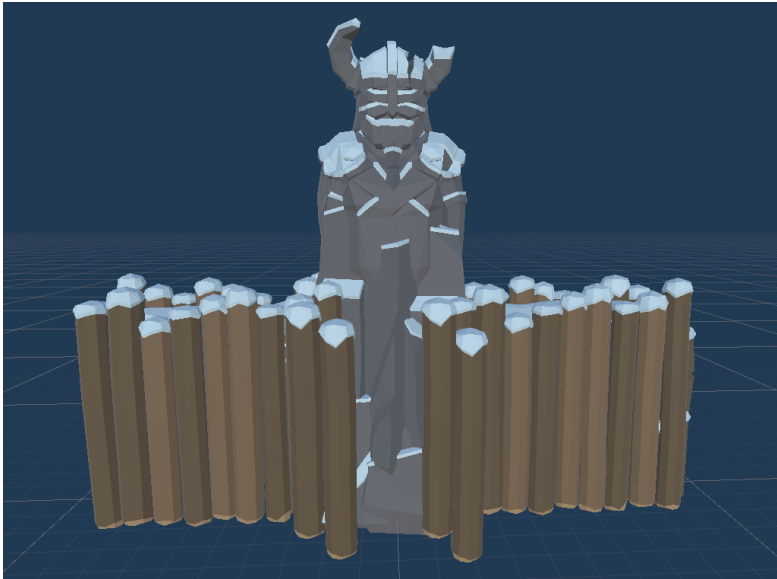


Figure 5.17: Wooden walls building design

Average cost of construction. High HP and general defense. These buildings appear in the game. However, they cannot be built as they need a specific script to align the walls with the nearby wooden walls buildings and this script has not been implemented due to a lack of time.

Wooden towers

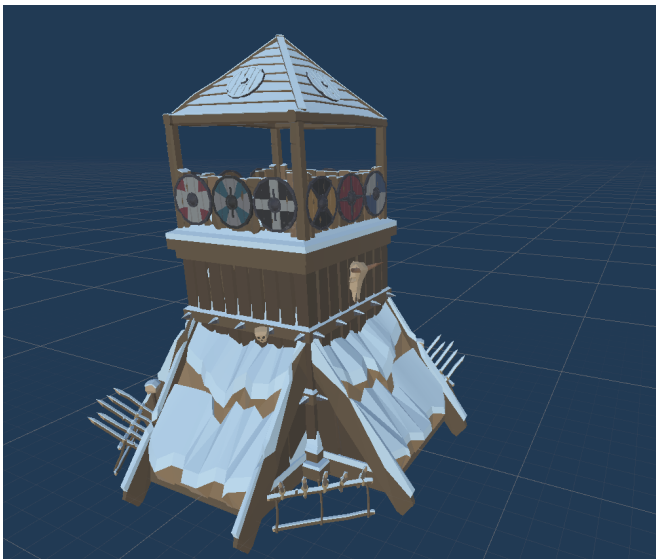


Figure 5.18: Wooden towers building design

Shoot nearby enemies in its area. High cost of construction. High HP and general defense. It has the additional properties of damage, range and attack frequency. Attacks automatically nearby enemies.

Stone walls



Figure 5.19: Stone walls building design

High cost of construction. Very high HP and general defense. As the player is not able to collect stone, the player is not allowed to build these buildings.

Stone towers



Figure 5.20: Stone towers building design

Shoot nearby enemies in its area. Very high cost of construction. Very high HP and general defense. It has the additional properties of damage, range and attack frequency. Attacks automatically nearby enemies. As the player is not able to collect stone, the player is not allowed to build these buildings.

Spawn building

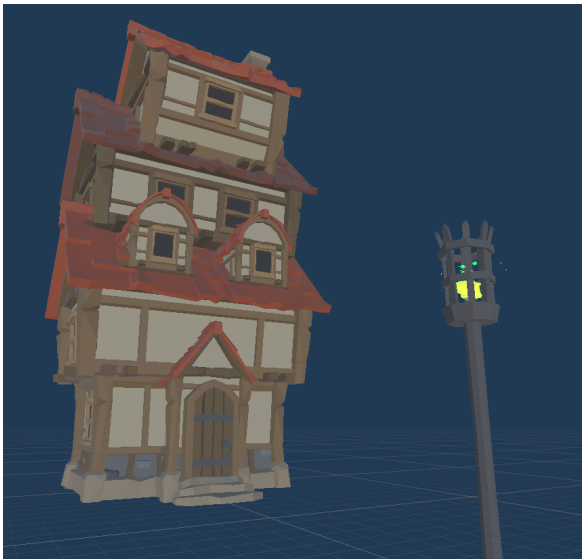


Figure 5.21: Spawn building design

Non-constructible building. Spawn troops nearby in a given position every space of time. The design of the building was already created in the asset.

Lord's castle



Figure 5.22: Lord's castle building design

It is a subclass of spawn building. Objective to be destroyed in every level.

5.1.7 Dynamic User Interface

Except for the hexagonal grid, the user interface (UI) was developed after the buildings and the troops. The 7 panels from which it is composed are controlled by a "UI Manager" script. These panels manage its shown information by a "Panel Manager" script, a script that I was able to create that is so generic that I can implement it in practically any other game. The "UI Manager" interacts with the "Level Manager" in order to get its information.

The construction of the UI took much more than expected, as I highly underestimated the task.

The icons of all UI are provided from Rexard's icons [16].

5.1.8 Level Manager

This element was not in the original plan of the project, but it was necessary from the start to the functioning of the game. It is the way abstract the forms of interaction, for example, it is inconvenient that every troop needed to store and manage the information panels of the UI in order to give its current attributes, that is why the troop stores the level manager and gives all of its information to, leaving the level manager the tasks of giving this information to the 'UI Manager'.

This component also manages the resources of the player, by removing the necessary resources when performing an action and collecting resources from buildings.

Additionally, this part of the project controls the state of the game by counting how many Settlers centers are in the Scene and how many Lord's Castles are. This way, when one of the Settlers Centers is destroyed and there are no more Settlers Centers, the game pauses and the players lose the game and a panel with a red background appears with:

- A title with the word "Defeat"
- A button to restart the game
- A button to exiting the game

When the player destroys a Lord's Castle, if there are no more enemy buildings of this type the game pauses and the same panel this time with a blue background and a "Victory" title appears.

5.1.9 Double interaction buttons

The element started its implementation during the development of the UI, due to the necessity to give more control to the player in the panel of the available buildings to be constructed and the panel of the available troops to be created. As described in the "Requirements analysis", these buttons perform two different actions depending on how many times in a row the button has been interacted with. In the case of this project, this implies that the first interaction shows adequate information in panels and the second performs the action assigned to the button. For its functioning to be possible, the buttons are managed by a "ClickIconManager" that controls the number of interactions in a row which each button.

5.1.10 Camera movement

After developing all pieces of the project above I programmed the movement of the Camera, which is not as easy as it seems at first glance, mainly because of the new unity's input system. Fixing the camera to a certain distance from the ground was easy, but the new input system gives a lower level of control over the inputs of the player and, because it was recently implemented in unity, the information on how to manage its input and control game elements with it is very limited. This meant that, in the end, I had to use my understandings of this system, and the one that I searched, and find a way to adapt old examples of camera control on mobile devices to the new system.

5.1.11 Enemy Artificial Intelligence

The next thing that followed developments was the enemy's artificial intelligence (AI). The enemy AI is very simple. The enemy starts with a set of builds with the only action that they perform is to spawn troops after a given time has passed and between a given amount of times. All of these troops have a "Puppet" script attached.

This "Puppet" script stores the positions of the "Settlers Centers" of the player and sends the troops that these objects are attached to the position where these buildings are located. As all the enemy troops have set the "Advance and attack" attribute, the troop will attack any enemy that may be found in the way. If the moment of the troop has been interrupted, usually because of combat, the "Puppet" script will recognize it and will set the troop destination again to one of the "Settlers center".

5.1.12 Game Screens

As I described in the “Requirements Analysis”, only the first game level is essential to the prototype, which has led to being this level the only screen of the project, even though it is true that the rest of the screen was not marked as optional as they were relevant to the game as a whole.

5.1.13 First Level

The last task that I have managed to do is to create the first level. It is not as polished and complete as I would have liked to, but it does a good job presenting the game even though I could not implement a coastline and the level does have more hills than what I originally planned.

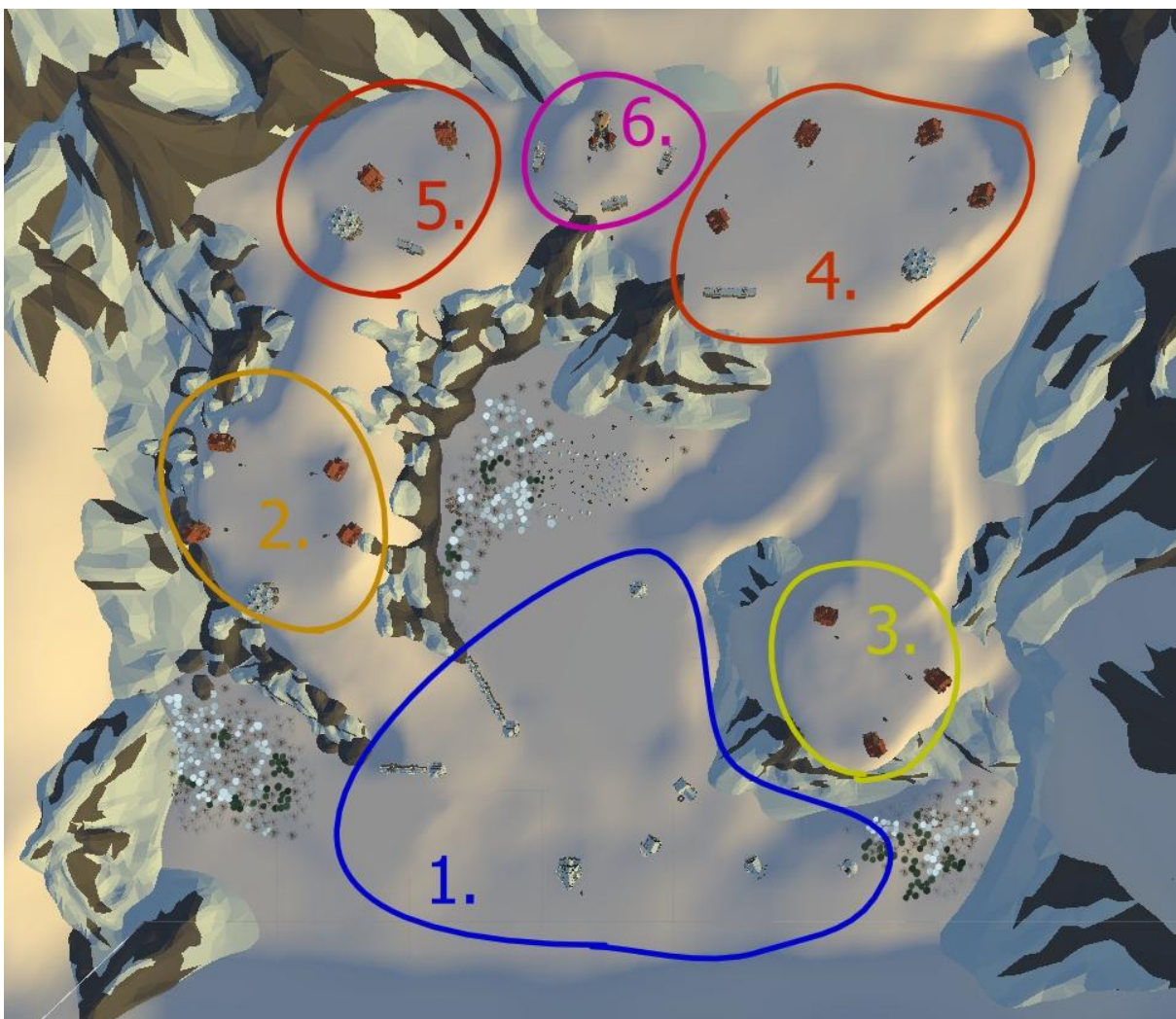


Figure 5.23: Top view from the first level and sectors separation

The level is divided into two types of areas, forest and hills, with 6 sectors in total:

- Sector 1: Sector where the players start. It has at least one building of every constructible building for the player and some wooden walls.

- Sector 2: Enemy outpost sector that will attack first. It has 4 buildings and one stone tower.
- Sector 3: Enemy outpost sector that will start sending attack groups to the player no longer after Sector 2 has started doing it. It has 3 spawn buildings.
- Sector 4: Enemy outpost sector that will start sending attack groups to the player no longer after Sector 3 has started doing it. It has 4 spawn buildings, a stone tower and a stone wall.
- Sector 5: Sector where the Lord's Castle is located. It does have some stone walls and spawns more troops than a regular spawn building. It starts sending troops to attack at the same time as Sector 4.
- Sector 6: Enemy outpost sector that is the last on starting sending attack groups to the player. It has 2 spawn buildings, one stone tower and one stone wall.

Also, so the player cannot construct buildings on inaccessible areas, I had to modify a part of the Hex Grid code. This modification consisted of forcing the script to not to create hex cells in places where invisible objects called "Selection Blockers" are placed.



Figure 5.24: Example of use of the hex grid code modification

5.1.14 Sound

Due to lack of time and the scale of the project, I was unable to implement audio in the game.

5.2 Results

Even though there have been a lot of changes to the plan and some of the features could not be implemented, all of the objectives have been completed.

The 2 levels artificial intelligence that controls the troop and units behavior is not perfect but it does implement all the necessary functionalities.

The UI is intuitive and easy to use and only shows the needed information of the current situation of the game.

The building system lacks elements in the terrain with special resources for some of its buildings and some buildings were not implemented in the building system because they needed special behavior that could not be implemented. However, most of the buildings can be constructed and implemented and work properly.

The modular assets from the Synty Store were used in a way that gave the game its own personality and the project is differentiable from other games that use the same asset packs from this company.

There are different types of buildings and troops developed from the generic scripts generated for those, proving that my script is robust enough to carry many different entities for the game with different 3D models, attributes and behaviors.

The level is not as polished as I would like it to be and differs from the original concept, but it is still a solid level for the prototype.

And in the end, the most important of all, the game prototype shows properly the concept of the game and its potential, although I cannot deny that it is not as polished as I would like, my expectations at the beginning of the project were not realistic.

The builds for Windows and Android can be found here:

https://drive.google.com/drive/folders/1gLIKQNS36y9vKV3Jhn0AflTqDSV68_f?usp=sharing



Figure 5.25 Screenshot of combat

6 Conclusions and Future Work

In this chapter, the conclusions of the work, as well as its future extensions are shown.

6.1 Conclusions

During the 4 years of my career, I have had problems with overly ambitious projects that I imposed on myself. I thought that I could solve this problem by making a mobile game, but I always find a way to dream too big, and I did not realize the size of the project until I completed the GDD. However, I am surprised about how much I have archived with this project, how much I have learned and how far I have reached.

To put this in perspective I compare my work done with the game 'Bad North' [20] that was developed in a bit more of a year by a team of 3 professionals that have spent years working on other companies like Ubisoft. The fact that I reached to accomplish something that is even remotely close to what they have done makes me proud.

However, I will still try to put my best effort into creating more modest projects. I would really like to create, even if it is small, a personal and complete game.

6.2 Future work

I would not be surprised if I continue the prototype and turn it into a full game in the future. The 'hard' part of the project is already done, and from this point, the only thing that it has left to do is to polish more the game and give it more content. However, creating a good product would probably need another 3 or 4 months of around 4 to 6 hours of work per day for 5 days a week, something that I am not sure I want to do, at least now.

7 Bibliography

1. Jeff Atwood. Stack overflow. <https://es.stackoverflow.com/> . Accessed: 2021-07-1.
2. Google. Google documents. <https://docs.google.com/> . Accessed: 2021-07-1.
3. Unity Technologies. Unity. <https://unity.com/es> . Accessed: 2021-07-1.
4. Unity Technologies. Unity forum. <https://forum.unity.com/> . Accessed: 2021-07-1.
5. Jasper Flick. <https://catlikecoding.com/unity/tutorials/hex-map/> . Accessed: 2021-07-1.
6. Google. Youtube. <https://www.youtube.com/> . Accessed: 2021-07-1.
7. Unity. Unity manual. <https://docs.unity3d.com/Manual/index.html> . Accessed: 2021-07-1.
8. The Document Foundation. Libre office. <https://es.libreoffice.org/> . Accessed: 2021-02-22.
9. Adobe. Mixamo. <https://www.mixamo.com/> . Accessed: 2021-04-12.
10. diagrams.net. draw.io. <https://app.diagrams.net/> . Accessed: 2021-07-1.
11. Grammarly, Inc. Grammarly. <https://www.grammarly.com> . Accessed: 2021-07-1.
12. Synty Store. POLYGON - Knights Pack.
https://syntystore.com/products/polygon-knights-pack?_pos=1&_sid=a31e4806a&_ss=r . Accessed: 2021-02-27.
13. Synty Store. POLYGON - Adventure Pack.
https://syntystore.com/products/polygon-adventure-pack?_pos=2&_sid=a31e4806a&_ss=r . Accessed: 2021-02-27.
14. Synty Store. POLYGON - Vikings Pack.
https://syntystore.com/products/polygon-vikings-pack?_pos=1&_sid=952d84efc&_ss=r . Accessed: 2021-02-27.

15. Synty Store. POLYGON - Particle FX Pack (Unity Only).
https://syntystore.com/products/polygon-particle-fx-pack?_pos=1&_sid=3b38442cb&_ss=r. Accessed: 2021-02-27.
16. REXARD. Icons by Rexard. <https://assetstore.unity.com/publishers/13229> .
Accessed: 2021-02-27.
17. Ludiq. Peek - Editor Toolkit.
<https://assetstore.unity.com/packages/tools/utilities/peek-editor-toolkit-149410> .
Accessed: 2021-02-27.
18. Gantt Project. Ganttproject download archive. <https://www.ganttproject.biz/download> .
Accessed: 2021-06-25.
19. Supercell. Clash of clans. <https://supercell.com/en/games/clashofclans/> . Accessed:
2021-02-29.
20. Plausible Concept. Bad North. <https://www.badnorth.com/> . Accessed: 2021-02-29.
21. Abuksigun. Siege Up!. <https://abuksigun.com/> . Accessed: 2021-02-29.
22. Paladin Studios. Stormbound. <https://paladinstudios.com/stormbound/> . Accessed:
2021-02-29.
23. Synty Store. POLYGON - Samurai Pack.
<https://syntystore.com/products/polygon-samurai-pack> . Accessed: 2021-02-29.
24. doorfortyfour. Databox - Data editor & save solution.
<https://assetstore.unity.com/packages/tools/utilities/databox-data-editor-save-solution-155189> . Accessed: 2021-02-29.
25. The Creative Assembly and Feral Interactive. Total war: Warhammer.
<https://www.totalwar.com/games/warhammer/>. Accessed: 2021-02-30.
26. Ubisoft and Ubisoft Blue Byte. The Settlers V: Heritage of Kings.
https://store.ubi.com/es/game?pid=5be2b34288a7e3b8170265d7&dwvar_5be2b34288a7e3b8170265d7_Platform=pcdl&edition=Heritage%20of%20Kings%20-%20History%20Edition&source=detail . Accessed: 2021-02-30.
27. Valve. Steam.
https://store.steampowered.com/app/364360/Total_War_WARHAMMER/?l=spanish .
Accessed: 2021-02-30.

8 Other considerations

8.1 Source Code

Since the project is composed of around 30 classes and a lot of them have around 200 and 300 lines of code, I decided that it is better to leave a link in a folder with all the source code, the build of the project and the unity project.

Link:

<https://drive.google.com/drive/folders/1Loq1ISwvUhh73P4jQ6OGsSxJpKvoxn8B?usp=sharing>

8.2 List of Figures and tables

Figure 1.1: Screenshot of 'Bad North' [20] gameplay. Page: 7

Figure 1.2: Screenshot of 'Siege Up!' [21] trailer. Page: 7

Table 2.1. Tasks and time dedicated. Page 9
Figure 2.1: Gantt chart of the tasks (made with Gantt Project [18]). Page 10
Figure 3.1: Game flow chart summary. Page 13
Figure 3.2: Screen Flow Chart. Page 20
Figure 3.3: Visual representation of the user interface. Page 24
Table 3.1: Case of use «CU01. Select cell from the hex grid». Page 29
Table 3.2: Case of use «CU02. Select troop». Page 29
Table 3.3: Case of use «CU03. Select building». Page 30
Table 3.4: Case of use «CU04. Move troop». Page 30
Table 3.5: Case of use «CU05. Attack enemy». Page 31
Table 3.6: Case of use «CU06. Create troops». Page 32
Table 3.7: Case of use «CU01. Construct a building». Page 33
Table 3.8: Case of use «CU08. Move the camera». Page 34
Table 3.9: Case of use «CU09. Win the game». Page 34
Figure 4.1: Case use diagram (made in draw.io [10]). Page 35
Figure 4.2: UI of the Prototype. Page 36
Figure 5.1: Original “Hex Map” Script Behaviour. Page 38
Figure 5.2: My modification to “Hex Map” Script Behaviour. Page 39
Figure 5.3: Troops groups from “The Settlers V: Heritage of Kings” game. Page 40
Figure 5.4: Black Hole Sun troops in formation. Page 40
Figure 5.5: Total war Warhammer [26] troops (Screenshot taken from Steam [27] store).
Page 41
Figure 5.6: Thrasher troop design. Page 44
Figure 5.7: Hunter troop design. Page 45
Figure 5.8: Swordsman troop design. Page 45
Figure 5.9: Zombie troop design. Page 46
Figure 5.10: Knight troop design. Page 46
Figure 5.11: Settlers Center building design. Page 47
Figure 5.12: Port building design. Page 48
Figure 5.13: Sawmill building design. Page 49
Figure 5.14: Hunting shack building design. Page 49
Figure 5.15: Iron mines building design. Page 50
Figure 5.16: Forge building design. Page 50
Figure 5.17: Wooden walls building design. Page 51
Figure 5.18: Wooden towers building design. Page 51
Figure 5.19: Stone walls building design. Page 52
Figure 5.20: Stone towers building design. Page 52
Figure 5.21: Spawn building design. Page 53
Figure 5.22: Lord’s castle building design. Page 54
Figure 5.23: Top view from the first level and sector separation. Page 56
Figure 5.24: Example of use of the hex grid code modification. Page 57
Figure 5.25 Screenshot of combat. Page 58