**Design Plan**

# 1 - Scope

The following document outlines the fundamental functionality that can be expected from the project, specifying the gameplay mechanics that will be found when the prototype is finished, as well as the terminology that will be used. This primarily covers the theoretical concepts rather than the technical implementation that will be handled.

Any text highlighted in red suggests an element with lower priority to be implemented into the game during development. The scope is also divided into two parts, the basic functionality, which is the bare minimum that is expected, and the Aspirational Scope. This section outlines the basic functionality.

**Note that the contents are subject to change in the actual product.**

# 2 – Resources

Players are expected to micro-manage resource to facilitate the upkeep and development of constructing an army. Purchasing units is a process involved with resource management – what type of units a player will purchase, how many, and when are key processes involved, to elicit critical thinking on how to handle various scenarios.

* **Gold** – Standard currency used to purchase units – units can vary in power and thus so does their cost in gold. A fixed amount of gold is passively generated per turn when owning certain *‘structures’*, with a greater quantity of said *‘structures’* leading to more passive income.
* **Food** – A resource to limit the number of deployable troops on field. This limits the amount of force a player can exhibit at a given time, without directly weakening them. Food is a value that serves as a cap to prevent a player being too dominant by having too many troops. Food can be increased by increasing the number of *‘villages’* owned by a player up to a certain point (hard cap).   
  E.g., 7/10 would suggest the following - 7 units are owned by the player, with their food cap being at 10, meaning they can only purchase 3 more units at most.  
  Note: Units may have variable food costs in the future, but for now they all cost 1. Current hard cap is 50 food.
* **Lumber –** Another currency that is passively generated. Can be used in conjunction with gold to purchase specific units. Alternatively, can be used to repair damaged *‘structures’* owned by the player.

# 3 - Map

For the time being, Maps are pre-designed static environments, where both users will compete on. More maps will be developed depending on time constraints which the players can choose to select during matchmaking.

Maps are a grid of varying dimensions (e.g., 150 by 100 squares), with each tile hosting a ‘terrain’ – these terrains determine how units may interact with them.

## 3.1.1 - Terrain

Environment tiles that can influence units that move through or are deployed on.

* **Forest** – Hinders *‘Mounted’* unit movements. Grants a 10% defence bonus to unit deployed on here.
* **Mountain** – Cannot be traversed by *‘Mounted’* units. Grants a 25% defence bonus to unit deployed on here.
* **Village** – Does not influence movement. Unit deployed here passively regenerate 10% *‘Health’* on players’ turn.
* **River** – Cannot be traversed by *‘Mounted’* units and hinders *‘Foot’* units.
* **Swamp** – Hinders *‘Mounted’* unit movements.
* **Desert** – Hinders all unit movements.
* **Plains** – Does not influence movement.

## 3.1.2 – Structures

Buildings which occupy a tile and can be captured if not owned by a player via attacking it to deplete its health. *‘Structures’* offer various utility based on what type it is categorised as. They are as follows:

* **Keep** – The win condition, where players will aim to capture the enemy keep whilst defending their own from attacks. Passively generates +25 ‘gold’ per turn. Additionally, increases food count by 10. Specified units can be summoned by purchasing here. Only one is generated on the map per player. Units deployed on the *‘keep’* will gain a 50% *‘defence’* bonus.
* **Village** – Passively generates +25 ‘gold’ per turn. Additionally, increases food count by 5.
* **Gold Mine** - Passively generates +50 ‘gold’ per turn.
* **Barracks** – Structure used for purchasing and summoning troops via resources. More variations will be included as the unit pool expands, e.g., Archery Range, or Stables.
* **Lumber Mill** - Passively generates +50 ‘lumber’ per turn.
* **Fortress** – Provides a 20% increase in *‘attack’* & *‘defence’* for units within a 3-tile range.

Units can only be deployed on a structure if they are owned by the same faction, any will function like terrain, otherwise they act as enemy units and can be attacked by not deployed on. All *‘structures’* grant a 20% *‘health’* regeneration per turn for deployed units – this will cost a fixed sum of gold.

# 4 - Combat System

The following explains basic rules that exist within the combat system, as well as detailing key terminology that will be used to describe how certain elements interact and function.

4.1 - Parameters  
Basic pre-defined stats (numerical values) that each unit possess that defines the degree of influence a unit has when interacting/interacted with. They are as follows:

* **Attack** – The amount of damage a unit can ‘inflict’ on a target.
* **Defence** – The amount of damage that can be mitigated when attacked.
* **Health** – The amount of damage the unit can receive before they are removed from play. To keep things simple, **every** unit has a fixed health value of 25.
* **Movement** – The maximum number of tiles a unit can traverse, not considering terrain which limits or prevents movement.

## 4.2 - Attack/Defence Attributes

Armour + Weapon types are attributes attached to every unit and will help determine the damage coefficient during combat based on whether an attack if ‘effective’ or not – each unit will only possess one pre-defined armour or weapon type.

* **[Armour Types] –** *Unarmoured*, *Light Armour*, *Heavy Armour*, Fortification.
* **[Weapon Types]** – *Pierce*, *Slash*, *Blunt*, (*Siege*, *Magic)*.

Effectiveness of an attack between two units is determined by matching **‘armour’** and **‘weapon’** types based on a ‘modified’ weapon triangle system (rock, paper, scissors).

[*Unarmoured*/*Blunt* **>** *Light Armoured*/*Pierce* **>** *Heavy Armoured*/*Slash* **>** *Unarmoured*/*Blunt*]

Attacks that are ‘effective’ (left hand side of ‘**>**’) grant a **+50%** increase in damage, neutral has no damage bonus, with ‘ineffective’ attacks granting **-50%** decrease in damage.

**Example:**A Heavy Armoured knight having a Slash weapon type will deal no bonus damage against a similarly Heavy Armoured knight as it is neutral, however, will deal +50% extra damage against *a* Light Armoured archer due to being ‘effective’. The same Heavy Armoured knight will take +50% extra damage when attacked by a Blunt templar, however will receive -50% less damage from a Pierce type archer.

Outside the triangle, exist extra types such as *Siege*, and *Magic*:

* Fortification receives -50% decrease in damage from *‘Blunt’*, *‘Pierce’*, and *‘Slash’* type attacks, but receives +50% increase in damage from *‘Siege’* type, and neutral from *‘Magic’*. All structures are *‘Fortification’* type, and some special units may receive.
* Siege is treated as a ‘*Blunt’* weapon type when attacking non-fortification units/structures, otherwise it deals +50% against ‘*Fortification’* archetypes.
* Magic does neutral damage to EVERYTHING.

## 4.3 - Unit Archetype

Units can fall into several archetypes which defines various factors regarding how a unit acts, outside of just combat.

**Attack Archetype** – Method of how the unit attacks:

* **Melee** – Unit must be directly one tiles away (right next to/neighbouring) to be able to attack.

|  |  |  |
| --- | --- | --- |
| X | X |  |

* **Ranged** – Unit must be exactly two tiles away (one tile space in between units) to attack.

|  |  |  |
| --- | --- | --- |
| X |  | X |

**Movement Archetype** – Method of how the unit movements:

* **Foot** – Can traverse all terrains but has the lower movement range.
* **Mounted** – Cannot traverse ‘Mountain’ and ‘River’ terrain, and has reduced movement for ‘Forest’, ‘Desert’, and ‘Swamp’ terrain, but has high movement range.

# 5 - How Combat and Damage Works

## 5.1 - Movement

Player owned units can be interacted with by selecting them and moving them to a tile within the scope of their maximum movement value. Units can move less than their maximum value or not move at all if the player so wishes. The movement can be inhibited by comparing the unit movement archetype and the terrain that they wish to travel through or upon.

The scope of a units’ movement is calculated by taking a units’ movement value and incrementally subtracting the *‘movement cost’* of the terrain (tile) the unit wishes to move through until it reaches 0 or below. By default, all terrain that does not hinder the units’ movement is 1, with terrain that inhibits movement being 2, and untraversable terrain being a roadblock. For example, a *‘mounted’* unit with a movement value of 7 would be able to traverse through x7 *‘plains’* terrain until it stops as:   
((((((7 – 1) – 1) – 1) – 1) – 1) – 1) – 1 = 0.   
Conversely it can also travel through x2 *‘plains’* and x3 *‘forest’* both stopping.  
((((7 – 1) – 1) – 2) – 2) - 2 = -1

**Case Study 1:**

A picture containing shape

Description automatically generatedA screenshot of a game

Description automatically generated with medium confidence  
The diagram above shows an example of movement of a unit interacting with terrain. The unit in this scenario is a *‘mounted’* movement archetype with a movement parameter value of 4 – meaning at most 4 tiles before stopping.

First let’s assume all the terrain is plains, meaning the *‘mounted’* unit isn’t affected when moving – the unit would be able to traverse and stop anywhere within the diamond of **[A5, E1, E9, I5]**.

However, a *‘mounted’* unit **cannot** traverse through or onto *‘mountain’* terrain and suffers a movement penalty from *‘forest’* terrain.

The *‘mountain’* terrain at [E7] prevents the unit from travelling directly down to [E9], causing [E6] to be the maximum range it can travel if going directly downwards. The unit can alternatively take another path to reach [E9], however, it will need to take a path *around* [E7], and due to the unit having only 4 movement, it can never reach [E9] in a single turn, no matter what route it takes.

The *‘forest’* terrain is less of a movement issue in this case study due to being on the outskirts of the maximum movement range. [F2, G3] are part of the original maximum movement range by default without considering terrain – the only reason [F3] is also considered such; is if the unit for some reason chooses to use [F3] as part of its routing, otherwise it can navigate around the *‘forest’*.

**Case Study 2:**

A picture containing shape

Description automatically generatedA screenshot of a game

Description automatically generated with low confidence  
*‘Forest’* becomes more of an issue in this scenario when travelling to the right, as the maximum movement range becomes more restricted due to the increased movement cost imposed by the terrain.

## 5.2 - Initiating Combat

A unit can attack a target *if* they are within during their movement. A unit being within attack range is determined based on the unit movement value and the units’ attack archetype which will determine how far they can attack. *‘Melee’* will allow the unit to attack a target directly one tile away from its end position, whilst *‘ranged’* will **only** allow it to attack a target 2 tiles away, meaning it cannot attack a target within melee range unless specified.

**Example:**  
Using the diagram for Case Study 2, if a *‘melee’* unit moves to G5, it can only attack if there is a target within [F5, G4, G6, F5]. For ‘ranged’ the attack scope is [E5, F4, F6, G3, G7, H4, H6, I5] assuming there is a target within any of the mentioned co-ordinates.

**Note:**  
After initiating combat and inflicting damage to the target, the enemy will immediately *‘counterattack’* if not killed. *‘Ranged’*, units cannot *‘counterattack’* against *‘melee’* units unless specified.

## 5.3 - Damage Calculations

Prior to calculating damage, effectiveness bonus/penalties are applied first to the attack parameter. If an attack is deemed to be effective based on comparing unit weapon type to enemy unit armour type, the unit will increase the attack parameter value by **+50%**, likewise, ineffective attacks will reduce the value by **-50%**.

Once bonuses and penalties are applied the attack occurs, and the damage inflicted on the enemy is calculated by with the formula:

**Damage = Attack Parameter Value - Enemy Defence Value**

If the calculated value is 0 or below, the damage inflicted is defaulted to 1. Afterwards damage is subtracted away from the targeted enemy’s current health value.

If the enemy’s current health value is reduced to or below 0, the enemy unit is ‘killed’ off. However, if not, the enemy initiates a ‘counterattack’ if possible – this means your unit will be receive damage using the same calculations including any bonuses and penalties.

**Note:**

Damage float values will be rounded to the nearest whole integer.  
Damage is dealt is fixed, there is no variance or RNG involved with this process.