# Phase 1

Team Fractal

# Contents

# Section 1

# Requirements

## 1.1 Background

## 1.2 User Stories

| # | Title | Story | Functional | Non-functional |
|---|---|---|---|---|
| 1 | GUI | As a player I must be able to see a GUI consisting of a map subdivided into plots and should be able to gain information about the state of my freehold and my individual plots. | 1.1.1. The entire map should be available to the user<br>1.1.2. Information about individual plots should be shown:<br>(a) Which player owns a plot<br>(b) Output of ore, energy, and food<br>(c) Robiticons installed | 1.2.1. The map must represent the university of York with at least 3 identifiable landmarks<br>1.2.2. The map must be split into multiple evenly sized plots<br>1.2.3. Each player must be uniquely identifiable on the map |
| 2 | Purchasing Land | As a player I must be able to purchase plots of land to increase the size and productivity of my freehold | 2.1.1. The player must be able to exchange currency for more land during the acquisition phase of the round | 2.2.1. Plots will have different strengths and weaknesses in terms of production based on location and terrain type<br>2.2.2. These values can change when random events occur<br>2.2.3. All plots are unallocated at the start of the game |
| 3 | Plot Modification | As a player, I must be able to buy and sell various modifications to my plots to increase productivity and/or style. | 3.1.1. The system must provide a number of possible modifications to plots | |
| 4 | Multiplayer | As a player, I must be able to buy and sell various modifications to my plots to increase productivity and/or style. | 4.1.1. A player must be able to chose whether to play against another human or the computer<br>4.1.2. At least two users must be able to play the game together<br>4.1.3. The players will take turns in playing | |

| # | Title | Story | Functional | Non-functional |
|---|---|---|---|---|
| 5 | Round Structure | As a player I must be able to play the game in a structured manner | 5.1.1. The game must be split into multiple rounds<br>5.1.2. Each round should be made of 5 phases:<br>1. Purchase any unoccupied plots<br>2. Purchase and customise roboticons<br>3. Install roboticons on plots of land<br>4. The colony produces resources<br>5. The player can buy and sell resources<br>5.1.3. Phases 2 & 3 must be time limited. | |
| 6 | Roboticons | As a player I must be able to purchase and customise my roboticons so they can produce more of certain amounts of resources | 6.1.1. The player must be able to purchase roboticons from the market<br>6.1.2. The market must have ore to produce roboticons<br>6.1.3. The user must be able to purchase modifications for the robiticon at the market<br>6.1.4. The user must be able to install modifications on roboticons<br>6.1.5. The user must have the option to install a roboticon on a plot of land they own. | 6.2.1. At the start of the game, the market has 12 roboticons |

| # | Title | Story | Functional | Non-functional |
|---|---|---|---|---|
| 7 | Resources | As a player I must be able to produce resources from my plots | 7.1.1. Roboticons are required to produce resources<br>7.1.2. During phase 4 the users roboticons will generate resources across the freehold<br>7.1.3. Food, energy and ore will be generated<br>7.1.4. Different amount of resources will affect the rate of production | |
| 8 | Buying/selling resources | As a player, I must be able to buy and sell resources to other players through an auction, or to the market at a fixed price so that I can maximise my wealth and productivity. | 8.1.1. The system must provide an auction facility, where the other player and the market bid for resources<br>8.1.2. The system must choose a market price based on resource abundance<br>8.1.3. The player must be able to buy/sell resources from/to other players, or the market | 8.2.1. At the start of the game, the market must have 16 units of food and energy and 0 units of ore<br>8.2.2. At the start of the game, the player must have a small amount of money |
| 9 | Gambling | As a player, I must be able to enter the bar and either win or lose money. | 9.1.1. The system must provide a minigame where the player can gamble with their money | |
| 10 | Winning | As a player, I must be able to win or lose the game. | 10.1.1. The system must assign a value to each resource at the end of the game, from which a player's final wealth is calculated<br>10.1.2. The game must end on the round in which the last plot of land has been allocated.<br>10.1.3. The player with the highest final wealth must be declared the winner, and Vice-Chancellor of the colony | |

# Section 2

# Architecture

## 2.1 Proposed Architecture

## 2.2 Justification

# Section 3

# Method Selection and Planning

## 3.1 Proposed Software Engineering Methods

hello [1]
goodbye [2, p 5]

## 3.2 Team Organisation

## 3.3 Project Plan

# Bibliography

[1] W. W. Royce, “Managing the development of large software systems,” 1970.

[2] “Rational unified process: Best practices for software development teams,” tech. rep., Rational, IBM, 1998.

# Section 4

# Risk Assessment and Mitigation

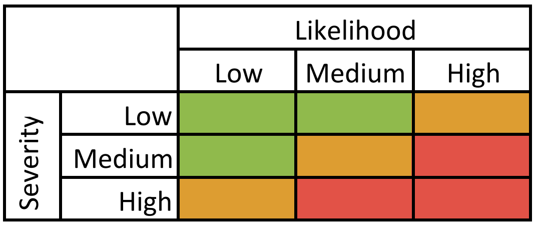
## 4.1 Introduction

Throughout the project we will face a number of potential risks however we will do our best to mitigate them. When analysing the risks we will focus on two factors, likelihood and severity.

A risk can either have a high, medium, or low chance of becoming a reality. If a problem is not likely to occur when running a project three times or more we deem it to have a low likelihood. If it is likely to occur once during the project or when running a project twice we deem it medium likelihood. Finally, if it is likely to occur multiple times during the project, we will deem it as high likelihood.

A risk can also either be high, medium, or low in severity. A high severity risk could result in months of lost progress up to having to totally start over. A medium severity risk could result in the loss of between one week and a few weeks of progress. Finally a low severity risk would only result in a maximum of a few days of lost progress.

To combine these two factors into something meaningful we will use a risk matrix (Figure 4.1). This will allow us to find a balance and identify the risks that will pose major problems.



| | | Likelihood | | |
|---|---|---|---|---|
| | | Low | Medium | High |
| Severity | Low | 🟩 | 🟩 | 🟧 |
| | Medium | 🟩 | 🟧 | 🟥 |
| | High | 🟧 | 🟥 | 🟥 |

Figure 4.1: Table showing how likelihood and severity of a risk combine to show overall impact of the risk

Green cells in the matrix are considered to be overall low risk, this is because they are not particularly likely to happen and if they do they will not have be severe enough to majorly impact the project. Orange cells signify an overall medium risk. They are either likely to happen but low severity, very unlikely to happen but would have a very severe impact or somewhere between. Overall high severity risks, red cells, are the most important to mitigate. They have a reasonably high chance of happening and could result in the loss of weeks or months of work.

It is important to categorise risks once they have been identified so that we can prioritise mitigation, it is imperative that overall high risks cannot happen and in the case that they do we must be able to cope with them and have protocols in place to lessen the impact.

We will be presenting the risks in a risk register with columns identifying, analysing and showing the mitigations for the risk. This will give us an accessible and easily modifiable document which we will be able to use throughout the project when considering or attempting to mitigate risks. [1]

## 4.2 Risk Assessment

# Bibliography

[1] B. Boehm and R. Turner, *Balancing agility and discipline: A guide for the perplexed.* Boston, MA, USA: Addison-Wesley Professional, 2003.