

1.0 Introduction

The first step we took when starting the requirements engineering process was gathering the requirements for the system, otherwise known as requirements elicitation. This involved collecting requirements from users, customers and other stakeholders. We followed the process described by Pierre Bourque and Richard E. Fairley in SWEBOK V3.0 – Guide to the Software Engineering Body of Knowledge [1]. The overall goal for the solution is to produce a playable and enjoyable game, which the customer can market and sell. A secondary goal for the project is for the team to achieve a good mark for the SEPR module. The stakeholders are the primary source of requirements as they are the people whom the product is for and ultimately will be assessing the solution, and hence making sure our product meets their specification is paramount to the team's success. There are two key stakeholders who are also sources for requirements, these being the customer of the project and the University of York Communications Officer to a lesser extent as we are not able to communicate directly to them.

The operational environment of the solution is something to consider when deciding on what technologies to use as this may impact how accessible our product is, for example with different operating systems or system architectures. This may influence the requirements for the project as some things may not be feasible within a certain environment. The other key source for requirements is the scenario document for the project as this outlines the task to be completed and also gives concrete and abstract requirements/constraints for the solution.

In order to come up with all the necessary requirements for this project, we started with analysing the scenario document and writing down as many points we could establish from this. This enabled us to outline any details we thought were too vague or unclear which we could then take to the client and, in an interview, ask them for clarity or direction with them.

When meeting with the client we started by asking directed questions in order to address some of the team's key concerns and this enabled us to get a better understanding of the overall scenario and end product. After we had clarified on the specifics, we had a more open ended conversation in which we were able to elaborate on more issues in a general sense and got the client to talk about their overall vision of the solution and how they thought it should work. The interview was particularly helpful in identifying areas that the client had specific requirements or direction for and those which were open to interpretation from our team.

From the interview phase we started developing use-cases and scenarios to better outline the specifics and requirements, starting with a high level use case for playing the overall game and then getting into more specific detail such as acquiring tiles and the in game economy. We used the use cases to produce the list of requirements as is outlined in this document based on how the game should function and how the user should be able to interact with the user interface.

The interview and discussion phase highlighted to us that the market interaction will be a big part of the gameplay characteristics and so a lot of thought and clear direction should be put into the development of it. The team realised that there are many ways in which this interaction could occur and so in order to make sure we were realising the client's visions we developed multiple paper prototypes for the market interfaces and demonstrated them to show how we thought it might work. From this we received feedback which we used to further refine the prototypes and ultimately produce a set of requirements for the market.

To form the requirements specification document we conducted research into the industry standards for software requirements documentation. The IEEE Software Requirements Specification [2] (SRS) provides an outline of the content and qualities of a good software requirements document. This specification organises the document into an overall description of the product and the functional or nonfunctional requirements, which includes use cases for the key features of the game. We modified this slightly to fit in with our own project as it is not being produced for use in industry and consequently we did not include some subsections which we deemed unnecessary. This includes the subsection for communications interfaces because there is no requirement for the game to run over a network. We also made the decision not to include a product perspective subsection in which we compare our game to similar products because we felt this was not necessary for the scale of our game.

1.1 Document Purpose

This document outlines the functional and nonfunctional/performance requirements of the game 'Roboticon Colony of York'. This specification is intended to give detailed requirements information for the customer to establish an agreement on what is expected from the product and for the development team to take further forward into the process. Therefore it is designed to prevent the project from failing by not meeting the customers needs.

1.2 Project Scope

The project utilises the team's skills to produce a colonisation game, The Roboticon Colony of York, that focuses on supply-and-demand economics and land acquisition on the University of York campus. The game will allow the player to produce, spend or auction their resources in order to beat their opponent. The player with the highest final score, determined by the resources held at the end of the game as well as other potential traits such as cumulative statistics, is pronounced the winner. The overall goal of this is to create a highly functional and enjoyable game that fulfills the customer's requirements.

Once this project is completed the game will be marketed and sold by the customer to users who wish to play the game. Ultimately the customer is who we need to convince of the validity of our game, hence we need to communicate often in order to fully understand their requirements and involve them in the decisions process. The University of York Communications Office is also interested in using the game for promotional activities such as Open Days and UCAS Days, meaning the game also needs to be of a high quality to help promote the university.

1.3 Definitions, Acronyms and Abbreviations

Term	Definition
Customer	The person/organisation that receives or uses the software product.
GUI	Graphical User Interface.
IEEE	Institute of Electrical and Electronics Engineers.
Roboticon	Robotic Assistant that can be customised to produce either Food, Energy or Ore.
SEPR	Software Engineering Project.
SRS	Software Requirements Specification.
Supply-and-Demand Economics	The interaction between the supply of a resource and the demand for that resource.
Use Case	A disciplined method of describing the typical behaviour of a component of the system.
User	The person operating and/or using the software system.
UCAS	The University and Colleges Admissions Service.

1.4 Document Overview

The remainder of this requirements document will be structured into two main sections:

- An overall description of the game which includes information about the logical characteristics of the different interfaces. This is split into four subsections: Product Functions, User Characteristics and Assumptions and Dependencies.
- A specific requirements section including the functional requirements of our game as well as any design constraints and nonfunctional requirements. This section is split into Software Product Features Requirements, Performance Requirements and Nonfunctional Requirements.

This structure follows the IEEE Software Requirements Specification [2], which lays out the functional and nonfunctional requirements and includes a set of use cases that describe user interactions that the software must provide. Use cases can be found on the website [3].

2.0 Overall Description

2.1.1 System Interfaces

The player will interact with the game using a system of GUIs. This will be set up in a clear and well organised form so that the user can understand and get full use from the game. The user interface and the map need to be entertaining and engaging.

2.1.2 User Interfaces

All interaction with the user is through the GUI. Menus will be interactive and easily accessible throughout the game. When the user is playing, everything they need will be clearly visible on the screen and easily accessible.

2.1.3 Hardware Interfaces

To run the game you will need a PC running any operating system. A functional keyboard and mouse is required to play the game.

2.1.4 Software Interfaces

The game will be capable of running on Windows.

2.2 Product Functions

The tile-based colonisation game will allow users to select plots of land so that they can produce resources. The game map will be of the University of York campus and will feature key landmarks to identify the location and help to market it to the University of York Communications office.

The main function of the game is to focus on supply-and-demand economics using the resources produced by the Roboticons. To allow the players to purchase and sell resources, the game will feature a marketplace with dynamic prices that vary depending on the availability of each resource. The market will function as a place where players can purchase Roboticons, trade resources or gamble money at the 'bar' in some sort of mini game. Players can interact with each other by purchasing each others resources in an auction-style process.

2.3 User Characteristics

There is one main user of our software, the player of the game. Therefore the player will need to have the following characteristics:

- Ability to read and understand English.
- Be able to use the controls including a mouse and keyboard.
- Familiarity with the operation of the basic GUI components of Windows or other operating systems.

No further abilities with computer technology will be assumed. If they cannot fulfill these characteristics, the user will not be able to play the game.

2.4 Assumptions and Dependencies

One assumption about the game is that it will always be running on a computer with enough performance to handle the software. It is also assumed that the computer will have the correct hardware resources including a working mouse and keyboard. The game is dependent on these hardware resources and therefore will not be able to function properly without them.

3.0 Specific Requirements

3.1 Software Product Features Requirements

The key product features of the game are the in-game economy and the market. Other features also include the purchasing and customising of the Roboticons, choosing a land tile and gambling.

3.1.1 Functional Requirements

The following functional requirements define functions of the game:

Requirement	Description
Multiplayer	The game needs to support at least two players, whether the opponent be human or computer. There is no requirement to support networked play but there is a risk that if we implement this it will be very time consuming. Use case 1 can be found on the website [3] and demonstrates how the game is played.
Plots of Land	The game incorporates plots of land, or tiles, which are uniform of size. These plots of land can have terrain features and can include buildings. Plots of lands can have different characteristics in terms of the resources present. Land tiles are free to obtain and limited to one per player per

	turn; demonstrated in use case 3 on the website [3].
Three resource types	The game features three resource types that are spread across tiles. These are Food, Energy and Ore. The user can collect these resources, with the winner being the player with the most amount of resources held at the end. Resources can be generated by placing Roboticons on land tiles.
A market players can interact with	A key feature of the game will be the marketplace where players can purchase and trade resources. The market will also include a 'bar' where players can gamble money within a mini game style setup. Use case 2 can be found on the website [3] and shows how users can interact with the market.
Gambling in the market	The market in the game will feature a 'bar' where players can gamble money by playing a game. This will be an exciting game that should entertain the user. Use case 4 demonstrates how the player can choose to gamble money in the market.
Trade	Players should be able to trade directly with each other or with the market. Use case 2 from the website [3] demonstrates how the user interacts with the market.
Dynamic Prices	The market focuses around supply-and-demand economics and prices are dynamic depending on the supply and demand for that product. See Use Case 2 on the website [3].
Customisable 'Roboticons'	Players can purchase Robotic Assistants, known as Roboticons. The player chooses whether this Roboticon produces either Food, Energy or Ore.
Timed rounds	Phases two and three of every round are time limited. Exact timings will be determined by play-testing.
Set amount of starting money	Players begin the game with the same set amount of money. This will be determined by play testing.
End game occurs when all tiles are taken	The game ends when no more plots of land can be acquired. The winner is determined by the player with the most resources.

3.3 Performance Requirements

Performance Requirement	Description
Tile information	Players should be able to view information about the land tile easily by clicking on the tile.
Using the market	The market should be easily accessible at all times. It should clearly incorporate a menu for buying Roboticons, trading and gambling. The market is centralised and not player specific.

3.4 Nonfunctional Requirements

Nonfunctional Requirement	Description
Dynamic storyline/narrative	The game should feature a dynamic storyline that relates to the University of York and makes the game more enjoyable. This is dynamic and changes throughout the game depending on the user's choices.
At least three visibly identifiable landmarks	There is no requirement that the game features the entirety of the University of York, but the customer requires that the game includes at least three visibly identifiable landmarks for marketing purposes.
Achievements	The game should include achievements that the player can collect. This will relate to the storyline and make the game more enjoyable and motivate the player to continue enjoying playing.

Bibliography

- [1] P. Bourque and R. Fairley, "Guide to the Software Engineering Body of Knowledge," IEEE, 2014, pp. 32-45.
- [2] IEEE, "IEEE Recommended Practice for Software Requirements Specifications," 20 October 1998. [Online]. Available: <http://ieeexplore.ieee.org/servlet/opac?punumber=5841>. [Accessed 25 October 2016].
- [3] SEPR, "Use Cases," 4 November 2016. [Online]. Available: <https://seprated.github.io/Assessment1/UseCases.pdf>. [Accessed 4 November 2016].