Updates Report

Any <mark>highlighted</mark> *parts of the individual updated documents indicate a change or addition from the original document in assessment 1*

Requirements

Original Requirements: https://github.com/jm179796/SEPR/blob/Assessment2_Docs/Assessment%201%20docs/Req1.pdf
Updated Requirements: https://github.com/jm179796/SEPR/blob/Assessment2_Docs/Updated%20Assessment%201%20docs/Req2.pdf

Some system requirements were rephrased and now focus on what functions the system will provide, rather than stating how those functionalities should be implemented. Some user requirements were altered or removed when implementation began as easier or more appropriate methods were found.

The following requirements have been altered (bold text refers to requirement name as listed in official requirements documents. The names are consistent through both original and new documents):

- **1.c.ii** - The original requirement states that information about the market and roboticons should be displayed in a pop-up screen. The new requirement is more general, and has abstracted the approach of showing the information as a pop-up screen; the information will now be shown on the game screen so that it is easier to find and use.
- **6.a** - The original requirement states the nature of how random effects will be implemented, this has been refactored to consist of a less precise definition. Now it only states the existence of random events and how they can have different levels of impact on the game. The requirement is now easier to test.
- **9.a.ii** - The original requirement states that only plots adjacent to plots the player currently owns may be acquired, this was scrapped to allow all tiles to be claimed regardless of the previous progress made by players. This new approach disallows players being "blocked off" and unable to acquire any new tiles, thus halting progress of the game.
- **9b / 9c**- The requirement **9b** was ambiguous as to when roboticons could be upgraded; the requirement **9c** now states that they must be installed on a plot of land *before* being upgraded.
- **14** - Technical details from the original requirement were removed, the new requirement describes what functionality has to be implemented without saying how it has to be done. The requirement is now easier to test.
- **14.a.i** - Equivalent to requirement **9.a.ii**, therefore this requirement has been removed.
- **14.b.i** - The original requirement had water as a resource instead of energy, this was an error and has been corrected.
- **16.b** - The original requirement states that the market menu must be accessed via a button; given that the requirement for the market menu was changed **[1.c.ii]**, this requirement was also generalised by not specifying the use of a button to access the market information. This makes the market information more accessible.
- **17** - The original requirement explained how players will select tiles and deploy roboticons, including information about the control scheme and menus used to do it. The requirement has been rephrased and technical details describing the processes of those functions were removed. The new requirement simply states that players will select tiles and be able to plant roboticons onto them. This is now easier to test and a simpler target to meet.

Method and Planning

The first section of the document, where the agile model was justified, has been shortened to attempt to give a more direct answer.

The Assessment 3 plan has been adjusted. A list of instructions with associated priorities has been provided for sprints, including tasks for the scrum team and scrum master; this has also been done for sprint reviews and progress meetings. It is important to add this detail so that it is clear what each team member, of a specific team role, must do during sprints of assessment 3 and to ensure that tasks with a higher priority are completed first. The plan was designed to be more general, in terms of completing specific tasks, so that it allows room for replanning if something does not work out how it was intended to.

Four additions have been made to the list of project resources. Gradle and IntelliJ were used to implement the project after assessment 1. Pixelmator and Paint.NET were used to create graphics for the project.

- Gradle is a build automation system. It is used to support incremental builds by determining which parts of the build tree are up to date and prevents any tasks dependant on those parts of build from being re-executed.
- IntelliJ is a java integrated development environment. It is a convenient tool to edit Java program code, supporting integration with GitHub remote storage and lets us link a Gradle project to an existing IntelliJ project.
- Pixelmator (Mac only), image editing software with the ability to do Pixel drawing and export to all the file types we needed.
- Paint.NET is an image editing software for windows. This tool was useful when developing the map for the game screen as it allows images with multiple layers (for buildings, land, tile grid etc).

New list of methods and tools that were used for this project and their web pages:

Methods and Tools	Web page
Agile	http://agilemanifesto.org/
Scrum	https://www.scrumalliance.org/why-scrum
GitHub	https://github.com/
Google Drive	https://drive.google.com/
ZenHub	https://www.zenhub.com/
Slack	https://slack.com/
LucidChart	https://www.lucidchart.com/
Smartsheet	https://www.smartsheet.com/
Travis CI	https://travis-ci.com/
Gradle	https://gradle.org/
IntelliJ	https://www.jetbrains.com/idea/
Paint.NET	http://www.getpaint.net
Pixelmator	http://www.pixelmator.com/mac/

Risk Assessment and Mitigation

Original Risk Assessment Document: https://github.com/jm179796/SEPR/blob/Assessment1_Docs/Risk1.pdf
Updated Risk Assessment Document:
https://github.com/jm179796/SEPR/blob/Assessment2_Docs/Updated%20Assessment%201%20docs/Risk2.pdf

The following changes were made to the Risk Assessment and Mitigation document (bold numbers refer to risk number as listed in new risk documents, other [numbers] refer to items at bottom of document):

- Added justification for risk rating scheme.
- Added number to each risk, so it is more convenient to refer to them.
- Categorised risks by adding a dimension to show which aspect of a project it covers, these consist of; requirements, project complexity, planning and control, team, and organisational environment [1]. These categories show the spread of risks across all areas of the project and assists with assignment of risk ownership.
- Risk ownership added. It was decided that the team leader is responsible for the risks related to the organisation and function of the group. The scrum master is responsible for handling the risks related to software development work. The remaining risks are divided between remaining team members (listed as "other" in the document).
- Added new risk **[26]** that states 'something is not working as planned in a sprint plan', this covers a situation where some unexpected issue arises; the solving of that issue should be accounted for in the sprint plan.
- Merged risks stating 'inadequate architecture, performance and quality' and 'final build has low quality' they describe very similar situations and therefore the latter has been removed, the former is now listed as risk **4**.
- Merged risks stating 'members of the team lack specialised skills required by the project' and 'the team lacks skills for the project and it leads to low productivity', the two risks are very similar and can be listed as one risk that is more general, the mitigation techniques were merged to cover all aspects of both risks. Now the risk can be identified more easily. It is now listed as risk **7**.

References

[1] Tharwon Arnuphaptrairong "Top Ten Lists of Software Project Risks : Evidence from the Literature Survey" International MultiConference of Engineering and Computer Scientists 2011 Vol 1, IMECS 2011, March 16 - 18, 2011, Hong Kong. [Online]. Available:
http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp732-737.pdf