

User Requirements

1. The game will not be convoluted and complex to play

- a. The game's element of challenge should be limited and aimed towards entry level players rather than masters of the game
 - i. Both our user and audience enjoy playing games to relax [1.2.1][4.3.1]
 - ii. It will need to be suitable for play by prospective students and other members of the public [3]
 - iii. The game should teach via gameplay and not have a tutorial or manual as a barrier to play the game, otherwise it won't be suitable for the intended environment. [1.8.1]
 - iv. The computer opponent should be easy to win against [1.1.1.4]

Risk: game could become too easy to win against computer opponent [R1]

Risk: game could be too hard to win against

Alternative/mitigation: If the game is too hard the computer player should be changed to be too easy, the game being too easy fits closer to the brief of the game being used on open days. This purpose is more suited to an easy demo than a hard one

- b. The timeframe for the game should be 10-20 minutes
 - i. Players will only be allowed to play the game in short sessions, so they must be able to complete it [1.4.1.2]
 - ii. Customer does not enjoy long games [1.7]

Risk: the game duration doesn't fit in this timeframe due to players taking too long/ being too quick when making decisions

Alternative/mitigation: The game being shorter than 10 minutes is acceptable provided it demonstrates every aspect of the game described below within that time. The game being above 30 minutes is indicative of poor game development for our project. Testing should be done as to why the games are lasting this long and the requirements re-developed to accommodate this unforeseen issue.
- c. The UI should be clear and beginner friendly
 - i. Text and UI graphics should be a reasonable size [7.4]
 - ii. The UI should link to other pop up screens for information like market prices and roboticon upgrades to promote a simple and structured experience [2.4]
 - iii. Only the current player's resource information should be visible [2.8]
 - iv. The statistics of tiles should be hidden until they can be acquired, so the user can apply intuition to their strategy [2.9]

Risk: the UI is too complex for the user to understand

Alternative/mitigation: Go back to designs, hold more prototype meetings with the customer and target the areas of the UI which are complicated to work out a more accessible solution

2. The game should have a graphical user interface which presents a map of the University.

- a. There must be at least three visibly identifiable landmarks
 - i. For university marketing purposes [3]
 - ii. Buildings may have unique bonuses for the player that occupies them [1.1.1.2, 1.1.1.3]
- b. The map should be subdivided into 16 isometric square plots of land (4 by 4) [1.8.2]
 - i. Plots may partly or totally include buildings
 - ii. Plots do not need to be identical in terms of resource allocation and buildings
 - iii. Square plots are favoured by the customer; they make the game easier to play for users [2.1]
- c. Plots must be all unallocated at the beginning of a game [3]

Risk: Graphical user interface is not detailed enough. It is hard to distinguish different objects. Icons are too small.

Alternative/mitigation: Refactor the resolution of the window to accommodate the GUI fully, ensure all graphics are distinguishable, create new graphics in house if required.

3. **The game should support two opposing players (one of which may be computer-controlled) [1.4.1]**
 - a. The user must be able to clearly distinguish between each player and their plots [1.5.1]
 - i. Each player should have different colour schemes to make the ownership of tiles clear [2.4.1]
 - a. Players should take alternating turns to play in a hot-seat style of gameplay [1.4.1]

Risk: Players may look at the screen during their opponent's turn and gain an advantage over them

Alternative/mitigation: Include in the tutorial a message informing players of good sportsmanship and the point of the game, remind them that 'screen peeking' is cheating and less fun.
4. **The game should take the general form of a resource-management simulator**
 - a. The game will have a scoring system with final score based off of resource amounts at the end of the game [3]

Risk: Resource-management logics implemented unrealistically. Some resources are too easy to gain and some are too difficult.

Alternative/mitigation: Either adjust the robots or the tile properties themselves to keep levels more realistic and less run-away.
5. **The player should have individual reserves of money and three different resources**
 - a. The resources are to be: Ore, Energy and Food [3]
 - b. Resources can be acquired, spent, sold, and bought (with players' money) [3]
 - c. Each player shall start with a small amount of money [3]
 - d. Player can win or lose money via "gambling" [3]
6. **The game should support random effects that come into play during a round [3]**
 - a. These effects should vary from low impact effects to larger more meaningful displacements in gameplay. Events should never be 'unfair' and only used judiciously. [1.6.1, 1.6.1.1]

Risk: Random effects may determine the outcome of the game.

Alternative/mitigation: Ensure any random effects don't determine phases they shouldn't, and that the effects for essentially a normal distribution not biased against one player or the other too heavily. As the customer said, employ randomness judiciously.
7. **The game should have a market which will be used to buy and sell resources and equipment**
 - a. The market will portray "Supply and demand economics" as described in [5,6]. [3]
 - i. The higher the amount of a resource in the market, the lower the price players will be able to sell that resource at.
 - b. The market will initially have 16 units of Food, 16 units of Energy and no Ore. [3]
 - c. The market will have 12 Roboticons (robotic assistants) at the start [3]
 - d. The market will have a bar where money can be won or lost via gambling [3]
8. **The player can purchase a robot ("Roboticon") to produce resources [3]**
 - a. Resources cannot be produced without Roboticons. [3]
 - b. Players can upgrade their Roboticon according to an upgrade path. [1.1.1.2, 1.1.1.3, 4.5.1.2, 2d)]
 - c. Roboticons are assigned to a tile. [3]
9. **The game should support rounds with 5 phases: [3]**
 - a. Acquisition: each player may acquire a previously unoccupied plot of land.
 - i. Will not include random elements [1.6.1]
 - ii. Only plots adjacent to plots the player currently owns may be acquired [2.3]
 - b. Purchase and Customisation: each player may purchase a Roboticon and customise them to support food, energy or ore production.
 - i. Customisation will cost money. [3]
 - ii. Phase will be time limited [3]
 - iii. Will not include random elements [1.6.1]
 - c. Installation of Roboticon: the customised Roboticon may be installed on a plot of land owned by the player.
 - i. Phase will be time limited [3]

- ii. Will not include random elements [1.6.1]
- d. Production: the players' colonies produce (or fails to produce) Food, Energy and Ore [3]
 - i. Suitable location for random effects applied judiciously [1.6.1]
- e. Auction: resources can be sold at an auction. Players may sell their resources to the other player or to the market at a fixed price. [3]
 - i. It will not be necessary for the player to sell or buy resources at each auction phase

Risk: Different phases are not introduced to player clearly enough and it could cause confusion.

Alternative/mitigation: Add a title screen to the phases, or use a ticker across the top of the screen, go back to the user with some sort of proof of concept for introducing the phases and making the current phase clear to the user to embed into the design.

Risk: phases may not be long enough for players to complete tasks in during allotted time

Alternative/mitigation: either set the phase timers to be too long, or carry out tests to judge average player action speeds

- 10. The game should end at the end of the round in which the last plot of land has been allocated
 - a. At the end of a game, the player with the highest score(4.a) is pronounced the winner [3]
- 11. The player can get information about his owned plots of land and roboticons installed on them
 - a. By clicking on the tile the player can get information about resources that it is generating [1.5.1]
 - b. By clicking on a roboticon the player can see the type of roboticon, it's current level and the number of resources that it produces [1.5.1]
 - i. The player can upgrade the roboticon using their money. The upgrade will enhance roboticon allowing it to generate more resources [1.1.1.3]

Risk: Information is not clearly represented

Alternative/mitigation: Go back to the user with any current efforts, get a detailed list of issues and problems and the user's ideas. Remake the information displays following this elicitation and return with a new design, repeat until all parties are happy.

- 12. The game should provide a brief background story
 - a. Customer and audience enjoy playing games with a narrative [1.3.1] [4.4.1]

Risk: It takes too long to review background story.

Alternative/mitigation: Make it skippable, or have a second much shorter version like a title screen. Make the context of the game clear through the game itself to add to the tone and the feel of the game rather than just a background story.

System Requirements

- 13. The main menu should offer the following options..
 - a. **Start Game**
 - i. This should bring the player to a menu where they can choose their college, whether they would like to play against a computer-controlled opponent or a human being, and an AI difficulty level (if applicable)
 - ii. Information regarding each college's unique attributes should be shown on this screen
 - iii. If the player chooses to play against someone else, another area of the UI will open up to allow for the other player to select their college
 - iv. Both players will not be able to select the same college to play as
 - b. **How to Play**
 - i. Brings up a flipbook-style sequence of pages that introduces the game's story/context to players and teaches them how to play it
 - ii. This should be concise and primarily convey information graphically [1.8.1.1]
 - c. **Leaderboard**

- i. Displays a list of the top 5 overall scores in the game and a simplistic name attached to each one

14. The game's map, which will be assembled manually and not through random generation, will be grid-like in nature and presented to players from a bird's-eye perspective

- a. **Players should be allowed to acquire tiles for their chosen colleges as the game goes on**
 - i. Only tiles that are adjacent to players' existing acquisitions will be acquirable
- b. **Tiles on the map will include:**
 - i. A value for Ore, Water and Food resources on that tile (possibly 0)
 - 1. Values unseen before acquiring
 - ii. At least three University departments and landmarks should be clearly visible[3]
 - iii. In accordance with 1.c.iv. The tiles resources should be representative of the graphic on them. I.e. A Lake will have a fair amount of water etc.

Risk: players may not be able to distinguish tiles

Alternative/mitigation: use artistic design effectively to represent differences in tiles using the map itself, or overlay a dotted grid of some kind over the map

c.

15. The player will interface with the game via the use of a mouse and keyboard

- a. Our game research showed that the mouse and keyboard combination is the much preferred method for playing any kind of strategy game due to the easier navigation and additional functionality [4.7.1]
- b. There will be keyboard shortcuts to access various menus quickly.

16. A heads-up display (HUD) should display the following information whilst playing

- a. Counters for current player's resources
- b. A button to access the market to, only show robot inventory in the market until auction phase [2.5]
- c. Button for the current user to end their turn
- d. A button in the corner which leads to the pause menu (19)

Risk: players may not understand elements of the HUD

Alternative/mitigation: use a combination of visual design, an in-game tutorial and tooltips if necessary to describe the HUD sufficiently

17. Players should be able to select tiles using the control scheme described in point [5] and plant available Roboticons onto them through tool-tip menus bound to each tile

- a. A tile's tooltip will describe/present the following (Provided they own it)
 - i. The type of tile that it is; The type of resources that can be gained from it; what kind of Roboticon is working on it (if one's working on the tile at all); options to assign or withdraw Roboticons from each tile and an option to upgrade the Roboticon working on the tile

18. Each tile should be visually distinguishable and follow a consistent art-style

19. Players should be able to pause the game at any time while it ensues

- a. Given that turns are timed, players should be able to attend to conversations or other tasks without it affecting the game.

20. The game needs to be playable on university computers

- a. It should not require a dedicated graphics accelerator other than those included with the processors powering the university's PCs
- b. It should not consume more than 4GB of RAM under any circumstances
- c. The combined file-size of the game and all of its resources should be small enough to fit on the university's servers comfortably
- d. Windows 10 should be supported as these are the operating systems that the university computers run
- e. It should not exceed a resolution of [1680 × 1050], as this is the resolution at which the monitors in the university's CS department operate

Risk: The specification of the university computers changes drastically during the timeframe of when the game is finished and when it is demonstrated

Alternative/mitigation: Refactor this requirement to comply with any new standards in the department