

Requirements (4)

*All changes have been made in red (#15, #16, #49, #50, #51), any additional requirements (not mentioned in other team's requirements but stated in initial brief) are in purple (#52, #53) and any requirements not implemented by previous team and us are in blue (#36).

We decided to adopt the requirements of the team whose project we chose (Duck Related Team Name). We decided to do this as the game we chose was modelled to these specific requirements, therefore we thought it would be easier to continue with the other team's requirements. However, we converted their requirements into a tabular form making them easier to read and assigned priorities and risks like we did with our original requirements.

| # | Category | Description | Risks | Priority |
|----|-------------------|---|--|----------|
| 1 | GUI | The system must ensure the entire map is available to the user | | 2 |
| 2 | GUI | The system should show the following information about individual plots: which player owns a plot, productions of ore,energy and food, roboticons installed | Incorrect information accidentally being displayed | 2 |
| 3 | GUI | The system must ensure the total amount of resources a user has is shown on the GUI | Incorrect information accidentally being displayed | 1 |
| 4 | GUI | The map must represent the University of York with at least 3 identifiable landmarks | | 3 |
| 5 | GUI | The map must be split into multiple evenly sized plots | | 2 |
| 6 | GUI | The system must ensure each player is uniquely identifiable on the map | N/A - Easy to implement | 3 |
| 7 | GUI | The system should ensure the GUI loads in less than 5 seconds | | 2 |
| 8 | Purchasing Land | The system must ensure the player is able to exchange currency for more land during the acquisition phase of the round | | 2 |
| 9 | Purchasing Land | The system should ensure plots have different strengths and weaknesses in terms of production based on location and terrain type | | 3 |
| 10 | Purchasing Land | The system must ensure the player is able to cancel a purchase to avoid accidental purchases | | 3 |
| 11 | Plot Modification | The system must provide a number of possible customisations to plots | | 2 |

| | | | | |
|----|-------------------|--|--|---|
| 12 | Plot Modification | The system must ensure the player is able to view all customisations and choose one to install | | 2 |
| 13 | Plot Modification | The system should ensure installation takes less than a second | | 2 |
| 14 | Multiplayer | The system should ensure a player is able to choose whether to play against another human or the computer | | 1 |
| 15 | Multiplayer | The system should ensure at least two users are able to play the game together. The game should support up to four players. | | 2 |
| 16 | Multiplayer | The system must ensure there is at least one human player | | 1 |
| 17 | Multiplayer | The system should ensure the players take turns in playing | | 2 |
| 18 | Multiplayer | The system should ensure the simulated player takes no longer than 20 seconds to complete a round | | 3 |
| 19 | Round Structure | The game must be split into multiple rounds | | 2 |
| 20 | Round Structure | Each round should be made of 5 phases: purchases unoccupied plots, purchase and customise roboticons, install roboticons, resource production and buy/sell resources | | 1 |
| 21 | Round Structure | The software must ensure phases 2 and 3 are time limited | | 2 |
| 22 | Round Structure | The system must ensure it is easy for the player to move between phases | | 2 |
| 23 | Round Structure | The system should ensure that changes between phases take no longer than 5 seconds | | 2 |
| 24 | Roboticons | The system must ensure the player is able to purchase roboticons from the market | | 1 |
| 25 | Roboticons | The system must ensure the market can only produce roboticons if it has ore | | 2 |
| 26 | Roboticons | The system must ensure the player is able to purchase modifications for the roboticon at the market | | 3 |
| 27 | Roboticons | The system must ensure the player is able to install modifications on roboticons | | 2 |
| 28 | Roboticons | The system must ensure the player has the | | 2 |

| | | | | |
|----|--------------------------|--|---|---|
| | | option to install a roboticon on a plot of land they own | | |
| 29 | Roboticons | The system must ensure the market has 12 roboticons at the start of the game | | 1 |
| 30 | Resources | The system should ensure roboticons are required to produce resources | | 2 |
| 31 | Resources | The system should ensure that during phase 4 the user's roboticons will generate resources across the freehold | The resource calculations may be performed incorrectly | 2 |
| 32 | Resources | The system must ensure that food,energy and ore are generated | | 1 |
| 33 | Resources | The system should ensure different amount of resources will affect the rate of production | | 2 |
| 34 | Resources | The system should ensure the resource production does not take more than 5 seconds | | 2 |
| 35 | Resources | The system should ensure the resource production happens automatically | | 2 |
| 36 | Buying/Selling Resources | The system must provide an auction facility, where the other player and the market bid for resources | | 3 |
| 37 | Buying/Selling Resources | The system must choose a market price based on resource abundance | | 3 |
| 38 | Buying/Selling Resources | The system must ensure the market has 16 units of food and energy and 0 units of ore at the start of the game | N/A - Easy to implement | 1 |
| 39 | Buying/Selling Resources | The system must ensure the player has a small amount of money at the start of the game | Balancing is an issue here, player needs to start with just the right amount of money to not be under/overtuned | 1 |
| 40 | Gambling | The system must provide a minigame where the player can gamble with their money | Need to balance to make gambling a risk, players should usually lose money and occasionally get lucky | 3 |
| 41 | Gambling | The system must ensure the minigame gives feedback on the money won or lost | | 3 |
| 42 | Winning | The system must assign a value to each | | 2 |

| | | | | |
|----|------------------|---|--|---|
| | | resource at the end of the game from which a player's final wealth is calculated | | |
| 43 | Winning | The game must end on the round in which the last plot of land has been allocated | N/A - Easy to track when no available plots left | 2 |
| 44 | Winning | The system must ensure the player with the highest final wealth is declared the winner and Vice Chancellor of the colony | Need to balance score to encourage multiple paths to victory | 2 |
| 45 | Random Effects | The software should ensure there is a chance each turn for an effect to influence gameplay that turn | Could become unbalanced very easily, playtesting will be needed to fine tune effects | 2 |
| 46 | Random Effects | The software should ensure either the player or player's plot(s) may be the target of random effects | | 2 |
| 47 | Random Effects | The system should ensure the effects are both positive and negative | | 2 |
| 48 | Random Effects | The system should ensure effects are clearly displayed so the player can tell exactly what has happened | | 3 |
| 49 | Additional Round | The game should have a 'capture the chancellor' mode | | 2 |
| 50 | Additional Round | The system should ensure the capture the chancellor mode should have a time limit of 15 seconds | | 3 |
| 51 | Additional Round | The system should award the player extra points if they capture the chancellor | | 2 |
| 52 | Environment | The game will be used on UCAS open days and other promotional events, so must be fit for display to the general public i.e professional and positive towards the university | | 2 |
| 53 | Environment | The aesthetic of the game should be futuristic and the setting of the game should be based in a futuristic University of York campus | | 3 |