

Requirements

The formal requirements are divided into three categories to improve readability and facilitate referencing. The categories are constraint requirements (external factors that affect the project), non-functional requirements (qualities that the game must have), and functional requirements (functionalities that the game must have). Each requirement is listed with its reference ID and any environmental assumptions, alternatives, and/or risks associated with it.

Constraint Requirements:

ID	Requirement	Environmental Assumptions	Risks	Alternatives
C1	The game must be completed and delivered by 7th May 2018.	All intermediate deliverable deadlines are met and an Agile process used.	With a definitive deadline, large late-project changes will be difficult.	None
C2	The game must not crash during any more than 1 in every 100 instances.	There is nothing wrong or lacking in the device running the system.	Difficult to obtain a sufficient number of tests to get an exact fail rate statistic.	Use thorough testing procedures to ensure reliability.
C3	The game must run at an average of 26 frames per second or better.	There is nothing wrong or lacking in the device running the system.	Framerate limit may be too high or too low, and should be revisited later in development.	Game should at least feel responsive to the user.
C4	The game must not cause any physical harm to the computer running it nor the user playing it.	The user is of required age, takes into account warnings provided, and uses game as instructed.	Possibility restraining game quality if harm to users is taken overboard.	None
C5	The executable for the game must not exceed 1GB.	There is nothing wrong or lacking in the device running the system.	Size limit may impede on the desired system outcome.	Game should at least be easy to distribute online.
C6	The game must run on Windows 10.	Majority of the targeted user base will own a device with Windows 10 (i.e. university computers).	All users may not have access to a device operating on Windows 10.	May expand to include Mac and Linux operating systems.

Non-functional Requirements:

ID	Requirement	Environmental Assumptions	Risks	Alternatives
N1	All code and documentation produced must be consistent, readable, and maintainable.	None	Could be difficult to enforce consistent coding conventions with multiple contributors.	None
N2	Design and implementation must be flexible to accommodate for any changes during development.	None	Changes in personnel, tools, and requirements could be problematic.	None
N3	There must be 4 players in each game.	At least 2 players must be controlled by a human player.	User may not want any computer-controlled players in the game.	Could select number of players at the start of the game.
N4	The game map must be based on the University of York campuses.	None	An accurate campus map may not be a balanced game map.	None
N5	The game map must be divided into sectors.	Must have at least 4 sectors. Sectors should be designed for games of about 20 minutes.	Poor distribution of sectors may create uneven game balance.	None
N6	Some sectors must be designated as landmarks. Each landmark must be associated with an amount of resources.	There must be at least 4 landmarks. Arrangement of landmarks should provide balanced gameplay. There cannot be more than one landmark per sector.	Too many landmarks that are close together may diminish their strategic worth.	Could remove resources if they become too complex to implement.
N7	Landmarks must correspond with real-life landmarks at the University of York.	Landmarks should be chosen carefully to maintain game balance.	Including all popular landmarks could create an unbalanced game map.	None
N8	The game map must offer a clear graphical representation of	None	Some common methods for distinguishing sectors	None

	sectors, unit position, and ownership of sectors.		(i.e. color coding) could reduce accessibility.	
N9	The outcome of a conflict must be random, but weighted according to the strength of the attacking unit compared to the strength of the defending unit.	If the attacking unit wins, the defending unit is destroyed and the attacking unit moves into the sector. If the defending unit wins, the attacking unit is destroyed.	Randomness may cause stronger units to lose conflicts to weaker units.	None
N10	In each turn, a player may choose to pass their turn.	None	Players may not want to pass their turn.	None
N11	The game should have a simple Main Menu.	The game must allow users to easily load a saved game or start a new game.	Implementing a main menu requires allocating additional resources to UI.	Add a "Load Game" button in the game HUD.
N12	The game must be easy for new players to interact with.	The game will be used at University open days and UCAS days, so inexperienced players should be able to interact with the game easily.	Dedicating much attention to user interaction may delay implementation of more important features.	None
N13	There must be different "levels" which a unit may be classified under. For example: 1st Year, 2nd Year, 3rd Year, 4th Year, and Postgrad	The level of the unit contributes to its strength in combat. Units should be able to increase their level over the course of a game.	Unit level could be difficult to clearly and concisely represent in a complex GUI.	None
N14	There must be at least 3 types of Punishment Card effects.	None	Certain card effects could be more powerful than others.	None

Functional Requirements:

ID	Requirement	Environmental Assumptions	Risks	Alternatives
F1	Any players not controlled by a human must be controlled by the computer.	A game may include 0 to 2 computer-controlled players.	Complexity of AI used to control players may cause performance and balancing issues.	Could use static, neutral units that are not controlled in any way and distributed randomly at the start of the game.
F2	A Pro-Vice Chancellor (PVC) must spawn in a randomly selected sector at the start of a new game. The PVC position should be kept hidden.	PVC should not spawn at a landmark or in a sector occupied by a unit. Only one PVC should be on the map at any time.	Capturing the PVC early in the game give one player an unfair advantage.	Could delay the PVC's spawn at the start of the game.
F3	When the sector the PVC is in is captured, a minigame should be played.	None	Could disrupt game rhythm if the minigame takes too long.	None
F4	Capturing the PVC should result in one or more players receiving a bonus of some kind.	The player who captures the PVC should always receive a bonus.	Nature of bonus could be too overpowered.	None
F5	Players must be able to capture sectors adjacent to those occupied by one of their units by moving a unit into them. Each sector may be owned by at most 1 player.	Each sector may contain at most 1 unit. Moving a unit into a sector already occupied by a hostile unit triggers a conflict. Capturing a landmark transfers the associated resources from the previous owner to the new owner.	None	None
F6	Players must receive new units.	Each unit spawns as a 1st Year. All units belong to 1 player. New units distributed to owned sectors.	Spawning new units too often could cause the map to become overcrowded and prolong the game.	None
F7	The game should be able to be paused, loaded, and saved.	None	None	None

F8	Players may use their Punishment Cards at any point during their turn.	Using a Punishment Card must impose some penalty on at least one other player.	Players using a large number of Punishment Cards at once could greatly affect the balance of a game.	Could set a limit on the number of Punishment Cards a player can use in a single turn.
----	--	--	--	--