Requirements

The requirements for the product were elicited through a process based on the methods outlined in SWEBOK Chapter 1 Section 3 [1]. The high-level goals of the system were provided in the scenario brief. In short, the system must be a multiplayer turn-based domination game that should provide an enjoyable experience to players and be suitable to showcase during university open days. Similar games in the genre, namely *Risk* and the *Civilization* series, were used as references for domain knowledge and informed parts of the initial design. The relevant stakeholders include the customer, the players, and the university recruitment office. The customer's primary objective was to create an enjoyable and marketable game in the specified genre. The primary objective of players was to have a positive experience with the game. The university recruitment office's primary objective was to show the game to prospective students during open days.

- *Team Discussion* - The basis for our requirements was determined initially via a team discussion of the scenario brief. Our discussion brought to light a number of questions that needed to be addressed in order to progress further in the elicitation process:
    - Some aspects of the scenario were not clearly defined. These included the allocation of sectors at the start of a game, the mechanics of the Pro-Vice Chancellor (the conditions to capture it, the minigame that follows its capture, and the type of bonus it would grant), the specifics of the bonus for capturing a sector, and the desired length of an average game.
    - Our discussion inspired ideas about additional features that might be added to the game to improve gameplay and player experience. We needed to ensure that these proposed additions were acceptable to the customer.
    - We also needed to know what content restrictions we would have to abide by. This was especially important for the purpose of showcasing the game during university open days.
- *Customer Meeting* - Our team arranged a meeting with the customer to obtain more information about the product, particularly about the objects of confusion mentioned above.
- *Use Case* - To ensure that every possible scenario in a game instance was accounted for, an exhaustive use case outline was created [link]. Once the outline was finalized, it was fleshed out to create a formal use case. This use case was used to elicit the requirements related to the specific interactions between users and the system.
- *Prototype* - A paper prototype [link] was constructed based on the working design, and a portion of a game was simulated. The prototype was used to highlight which requirements and design aspects worked well and which would likely need further tweaking later in development [2]. The prototype was also used to simulate a portion of a game for the customer to obtain approval for our design.

The elicitation process was carried out in a way that suited the size of our team and the scope of the project. An overly formalized and rigid approach would likely hinder the adaptability and flexibility inherent in small teams and projects. Instead, we created use cases and prototypes to identify the core requirements in an organic way, taking the perspective of the user into account [3].

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 sectors, unit position, and ownership of sectors.	None	Some common methods for distinguishing sectors (i.e. color coding) could reduce accessibility.	None
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

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.	All human players that have not yet been eliminated should participate	Could disrupt game rhythm if the minigame takes too long.	Could reduce the number of players participating.
F4	The player who captures the PVC and the player who wins the minigame should get a bonus.	If the player who captured the PVC also wins the minigame, the bonus should be larger.	Nature of bonus could be too overpowered.	Could have only the player who wins the minigame receive the bonus.
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. A sector can only be owned by one player at any one time. Capturing a landmark transfers the associated resources from the previous owner to the new owner.	None	None
F6	Players must receive new units each turn.	All units belong to 1 player. New units distributed to owned sectors.	Spawning new units each turn could cause the map to become overcrowded and prolong the game.	Time to receive new units could change to better balance the flow of the game.
F7	The game should be able to be paused, loaded, and saved.	None	None	None

References:

[1] P. Bourque and R. E. Fairley. (2014). *Guide to the Software Engineering Body of Knowledge* (version 3.0) [Online]. Available: https://www.computer.org/web/swebok/v3

[2] A List Apart. (2007, Jan. 23). *Paper Prototyping* [Online]. Available: https://alistapart.com/article/paperprototyping

[3] E. Kamsties, K. Hörmann, M. Schlich. “Requirements Engineering in Small and Medium Enterprises,” Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, Requirements Eng 3:84-90, 1998.