Requirements Elicitation

During the process of requirements elicitation we made several steps to ensure that the requirements we produced best matched what the customer intended for the product. The primary stakeholder in this case is the customer, with whom we directly conversed on several occasions. It was most important that the finalised requirements matched their vision of the game as a failure to achieve this would mark an overall failure of the project. A secondary stakeholder is the University of York Communications Office, who may wish to use the product in future to promote the university. Though we did not establish direct contact with the Communications Office, we found it advantageous to have a team member represent their interests during the process of requirements elicitation.

The initial list of user requirements was extracted from the brief provided by the customer. The user requirements were then discussed as a group and a list of questions [1] directed at the customer was produced in order to clarify any ambiguity or conflicts we perceived in these requirements and ascertain details that had not been explicitly provided in the brief. We organised an interview with the customer in order to address each of the questions on our list. Based on the customer's responses, we reviewed our list of user requirements, making adjustments and adding new ones as was appropriate. Several use cases [2], based on a template by A.Cockburn [3], were constructed in order to further clarify the ways in which certain features of the game will function, this was used to create new requirements as well as to modify and justify our existing ones. From the completed list of user requirements we extracted the appropriate system requirements associated with each, the fit criteria, and the relevant non-functional requirements. The customer then reviewed our finalised user and system requirements.

Whilst we were gathering our requirements we made several assumptions about the environment the software would be operating in. We assumed that the user would be using a keyboard and mouse to control the game, and a screen to view it. The user's computer is assumed to be no more than eight years old in terms of hardware and is expected to be running Windows 7 or above. The computer should be capable of executing Java code.

Requirements Presentation

We decided to use a modified version of the IEEE software requirements specification template [4] to present our requirements. Where the IEEE template lists system features, we have decided to instead list user requirements and then the associated system requirements in order to more clearly show the relationship between the two. We also decided a tabular layout would make it easier to locate requirements when referenced in later documentation as they are a lot more compact. Each requirement has a unique ID which denotes whether it is functional or non-functional as well as grouping related user and system requirements. The advantage of using a unique identifier for each is that it can be easily referred back to in any future documentation. Every system requirement also has a fit criteria which sets out the condition that, when satisfied, allows us to conclude that the requirement has been successfully implemented. The final column in the table provides the source(s) for each requirement as well as allowing for any additional non-functional requirements and environmental assumptions to be specified.

Requirements Elicitation

ID	User Requirement	ID	System Requirements	Fit Criteria	Justification, Assumptions, and associated NFR(s)
1	The amount of violence the user experiences in the game shall be minimal.	**NF1.1**	The game shall not display any graphic violence.	The customer shall be satisfied that the game is not too violent.	Extracted from the brief and discussed with the customer.
2	The user shall be able to save and pause the game at any time.	**F2.1**	The system shall store the current state of the game to a file when the user saves.	After saving a save file shall be present on the user's drive.	Extracted from the brief and developed from Use Case 2. This assumes that the game has appropriate file system permissions. **(NF2.4)** The save operation shall be completed in under 10 seconds.
		F2.2	Save files created should contain all information needed to reload a game.	A loaded save shall mirror the game at the point that save was created.	
		F2.3	Pausing the game shall pause the game timer and display a pause menu with basic options.	Pressing the pause button shall cause the game to pause as described.	
3	The game shall have a sector containing the PVC (Pro Vice Chancellor) and the user shall be awarded a bonus for capturing it. The PVC shall be a member of university staff.	**F3.1**	The map shall have one AI controlled sector containing the PVC (Pro Vice Chancellor).	Between the start of a game and the elimination of the neutral AI, the PVC shall be captured.	Extracted from the brief, discussed with the customer, and developed from Use Case 3.
		F3.2	The sector containing the PVC shall be randomly determined at the start of the game.	Multiple game loads shall produce different PVC locations.	
		F3.3	The location of the PVC is initially hidden and only revealed on first capture.	A new game map shall not indicate the PVC location.	
		F3.4	The system shall award a bonus for both initial and subsequent captures of the PVC.	All captures of the PVC sector shall trigger the award of a bonus.	
4	When the user captures the sector containing the PVC a mini-game shall be unlocked.	**F4.1**	The system shall include a mini-game, which is playable after the initial capture of the PVC.	Capturing the PVC shall unlock the mini-game.	Extracted from the brief, discussed with the customer, and developed from Use Case 3.
5	The user shall be able to choose which UoY college they want to play as.	**F5.1**	The system shall display a list of playable colleges before the game starts.	When a new game is being started, the list of colleges displayed shall match those the user can play as.	Extracted from the brief, discussed with the customer, and developed from Use Case 1.
		F5.2	The system shall allow the user to select which college they want to play as.	Selecting a college from this list shall start a game with the user playing as that college.	
		F5.3	The system shall represent different	Each in game college's logo	

			colleges in the game using identifiers such as logos and colours.	and colour shall match those of the real-world college.	
6	The game shall start with both players having an equal number of sectors and gang members. All sectors not under player control shall be controlled by the game until they are taken over by a player.	**F6.1**	At the start of each game the system shall randomly allocate sectors to the two players and a neutral AI.	Multiple game loads shall produce different sector allocations.	Extracted from the brief, discussed with the customer, and developed from Use Case 1. **(NF6.5)** The game shall take no longer than 30 seconds to generate and display the starting conditions of the map.
		F6.2	The neutral AI shall start with more sectors than the players.	At the start of the game the number of AI controlled sectors shall be greater than the number controlled by each player.	
		F6.3	The neutral AI shall not receive reinforcements.	The system shall never allocate new gang members to AI sectors.	
		F6.4	The neutral AI shall only defend its sectors.	The AI shall never attack a player controlled sector.	
7	The user shall be able to receive new gang members on each turn and also by converting collected bonuses.	**F7.1**	Each turn the system shall calculate how many new gang members to allocate to the player and then allocates those gang members.	The number of gang members allocated should match the calculated number of gang members to be allocated.	Extracted from the brief and discussed with the customer.
		F7.2	The system shall show the user the number of gang members they are eligible to receive based on the quantity of bonuses they have collected.	The number of gang members a user is eligible to receive shall match that calculated by the system based on bonuses.	
8	Each turn the user shall be able to end their turn by attacking a sector held by another player, moving within held sectors or passing.	**F8.1**	The system shall present the user with multiple options on each turn, though one of attack, move or pass is needed to end the turn.	The turn shall only progress if attack, move or pass is selected by the user.	Extracted from the brief, discussed with the customer, and developed from Use Case 1.
9	When the user attacks another sector, they should get control of that sector and a bonus if they win or potentially lose gang members if they lose.	**F9.1**	The system shall be able to calculate and compare the relative strengths of the opposing teams to calculate the outcome of the conflict when one team attacks a sector held by the other.	The outcome of conflict shall be reflective of the strengths of the opposing sides.	Extracted from the brief and discussed with the customer. **(NF9.5)** The system shall calculate and return the outcome of an attack within 2 seconds.
		F9.2	An attack shall be considered successful if all of the defending gang's members are lost in the conflict.	An attack where all of the defending gang members are lost shall trigger the relevant events, such as sector	

				reallocation.	
		F9.3	If an attack is unsuccessful, then one or both gangs shall see a loss of gang members but no sector control changes shall occur.	An attack where both teams are evenly matched shall result in both teams losing some gang members.	
		F9.4	When the user successfully attacks a sector, the system shall reallocate the sector to being under the user's control and the user shall receive a bonus at the end of their turn.	An attack where the player is successful shall result in the relevant bonuses being awarded and sector reallocation.	
10	The user shall be able to allocate new gang members to sectors under their control.	**F10.1**	The system will keep track of the number and affiliation of gang members in each sector.	The stored information relating to the number and affiliation of gang members shall match with what's displayed on the map.	Extracted from the brief and discussed with the customer.
		F10.2	The system shall allow users to add new gang members to sectors under their control and move gang members between them.	There shall be options that, when selected, shall result in a change in the position or number of gang members in sectors.	
11	The user should be able to view and interact with the map	**F11.1**	The game shall display a map of the UoY campus, split into a number of unequal sectors each representing a campus landmark, along with information about the current state of the game.	Correct information relating to the current state of the map and game shall be displayed to the user; this shall match the information stored by the system.	Extracted from the brief and discussed with the customer. The assumption is that the player is using a monitor to view the game, and a mouse and keyboard to interact with it. **(NF11.3)** Any delay when the user interacts with the map (or interface) shall be minimised.
		F11.2	The game shall provide an interface through which the user can interact with the game.	When running, the game shall display interfaces that match what is set out in the design documentation.	
12	The game shall end when only one gang remains, the turn limit has been reached (in which case the winner is the player with the most sectors at that point) or a player forfeits the game.	**F12.1**	When the system detects that the end condition has been met, the game shall end and a summary of the game statistics shall be displayed to the user.	Reaching an end condition shall trigger the ending events of the game.	Extracted from the brief, discussed with the customer, and developed from Use Case 1.

Bibliography

[1] Sid Meier, Initial Requirements and Customer Q&A. Available: https://drive.google.com/open?id=18Fev106YcyOHcYZrF0VSIoFuSB-QPg4K

[2] Sid Meier, Use Cases. Available: https://drive.google.com/open?id=1wbLzwvNHyga28qTMjW6JkD5zbRMe6vAL

[3] A. Cockburn, *Writing Effective Use Cases*. Addison-Wesley, (2001).

[4] K. Wiegers, *IEEE Software Requirements Specification Template*, (1999). Available: https://www.ce.yildiz.edu.tr/personal/kalipsiz/file/28827/IEEE830.pdf [Accessed 18th October 2017]