A)

We first met with the customer on 22/10/20 in order to elicit requirements. We went into the interview with a list of specific questions which the customer answered. They also passed on requirements given by the York university communications office (referred to throughout this document as YUCO). After the meeting we began to consider how we would structure our requirements. We split our requirements between our two stakeholders for clarity, and assigned them all priorities. Requirements that our stakeholders have stated are not negotiable have been assigned the priority “shall” while those that are important, but can be sacrificed if development should run past it’s time limit are designated “should.” Any other requirements are mere suggestions and are designated “may”. This priority system was implemented to guide development, so that those requirements which would cause the most inconvenience to the stakeholders if neglected, could be fulfilled.

We also researched the kind of language that should be used for this document, and our investigations led us to adopt a tone that is as unambiguous and specific as possible, lest an important requirement be incorrectly interpreted, and lead to something other than the intended functionality being implemented. At best, this would waste the time of the programmers on the project, as they would then have to remove the unnecessary code and reimplement the feature. In the worst case scenario, the mistake would go unnoticed and lead to the stakeholders being delivered deficient software.

Our research then led us to consider whether there were any non-functional requirements aside from the deadline and the 2D nature of the game. We posed questions regarding the accessibility support, but were told we did not have to consider needs beyond that of an average first-time user.

https://www.pjsrivastava.com/a-short-guide-to-writing-software-requirements
https://blog.testlodge.com/how-to-write-specifications-for-software/

User Requirements (YUCO refers to York University communications office.)

ID	Description	Priority	Stakeholder
UR_2D	The game should be top-down and 2D	Shall	Customer
UR_CONTROLS	The game should use mouse and keyboard controls	Shall	YUCO
UR_HARDWARE	Can run on a 13 inch laptop to a 32 inch desktop with scaling - else if YUCO attempts to run it on a desktop during open days, the window will be small, which may make it harder to play.	Shall	YUCO
UR_SCORE	Number on screen for how many imposters the player has arrested	Should	Customer
UR_MUTABLE	If music is implemented, it must be mutable	Shall	YUCO
UR_DEMO	Game has a demo “non-player” mode	Shall	YUCO
UR_SINGLE_PLAYER	The game is one player	Shall	Customer
UR_CONTROL_INFO	Game has a screen that informs the player how to use the controls.	Should	YUCO
UR_PG	The game must be PG.	Shall	YUCO
UR_WIN_STATE	The game must end and tell the user they have won if they arrest all infiltrators.	Shall	Customer
UR_LOSS_STATE_SABOTAGE	The game must end and tell the user they have lost if 15 systems have been sabotaged.	Shall	Customer
UR_LOSS_STATE_DEATH	The game must end and tell the user they have lost if they die.	Shall	Customer
UR_REAL_TIME	The game must be in real-time, not turn based.	Shall	Customer
UR_ROOM_VARIETY	There must be at least four types of rooms in the “station” that the game takes place in.	Shall	Customer
UR_TELEPORT_PADS	The rooms in the game can have teleportation pads which the player can teleport between.	Shall	Customer
UR_SPECIAL_ABILITIES	There must be at least 3 distinct special abilities between the infiltrators	Shall	Customer
UR_INFIRMARY	The player must be able to teleport to a room designated the “infirmary” where they can heal	Shall	Customer
UR_DEVELOPMENT_TIME	The game must be completed by the 23rd December 2020	Shall	Customer
UR_FIRST_TIME_USER	The game must be easy for a first-time user to learn.	Shall	YUCO

Why?

1. The ease of coding and also due to time constraints. Also, this makes it easier to draw/acquire textures and assets
2. Allows a user with no experience to learn the game quickly.
3. Our user intends the game to be used on open days - this will be a lot easier if it can be run on any device regardless of screen dimensions.
4. Allows the user to know how close they are to winning the game.
5. Due to the environment that the game will be presented in, any music will need to be mutable so as not to distract from other presentations in the room.
6. The environment the game will be presented in will require it to run a demonstration of itself while idle in order to attract players.
7. The game will not be on multiple screens, and therefore it would be impractical for multiple players
8. All our players will be first-time players, and therefore the controls of the game need to be easy to learn, thus providing instructions on how to use the controls is important.
9. Parents are likely to be in the environment the game will be presented in and may bring young children with them, therefore the game must be appropriate for all ages.
10. The user will not have long to spend playing the game, therefore it must definitively end.
11. See 10.
12. See 10.
13. The user is unlikely to have a lot of time to spend playing the game due to the environment it will be presented in, thus it must be in real-time and not be turn based, else it would last too long.
14. To allow gameplay to vary and prevent the user from getting bored.
15. The suggested teleport system allows the user to move around quickly, speeding up the pace of the game so that it will not last longer than the time the user will have the opportunity to play it in, and hopefully also making it more fun.
16. See 14.
17. Shortens the amount of time the player spends walking to rooms (see 13).

System Requirements

Functional Requirements

ID	Description	User Requirements	Fit Criteria	Priority
FR_MOVEMENT	When the user presses on WASD, the user's sprite should move on-screen.	UR_CONTROLS	The sprite will have X and Y class attributes that will be changed when the user users WASD.	Shall
FR_BOUNDARIES	The user shouldn't be able to move the player off the screen or out of the map	UR_CONTROLS	Keep the character within the boundaries of the map.	Shall
FR_SCORE	The number of infiltrators arrested should be	UR_SCORE	There should be a variable of type int. It should have one added to it whenever the	Shall

	displayed on the screen and update when the user arrests another.		arrest infiltrator function runs. It should be displayed on the screen at all times.	
FR_DEMO	The game should include a demo mode, during which the map is loaded and a preview of gameplay is shown.	UR_DEMO	The map should be rendered, and the player's sprite shown and moved around the map by the changing of it's X and Y position variables in a simulation of gameplay.	Shall
FR_WIN_STATE	The game should end when the player has arrested 8 infiltrators, and some text should be displayed on the screen to inform the user that they won.	UR_WIN_STATE	There should be a subroutine that checks if an int variable recording the number of infiltrators arrested **=8**. If so, the function that updates the game should stop running and text should be rendered on screen informing the user that they won.	Shall
FR_LOSS_STATE_SABOTAGE	The game should end when the player has arrested 8 infiltrators, and some text should be displayed on the screen to inform the user that they lost.	UR_LOSS_STATE_SABOTAGE	There should be a subroutine that checks if an int variable recording the number of systems sabotaged **=15**. If so, the function that updates the game should stop running and text should be rendered on screen informing the user that they lost.	Shall
FR_LOSS_STATE_DEATH	The game should end if the player dies and display text	UR_LOSS_STATE_DEATH	There should be a subroutine that checks if the int variable recording the player's heath **=0**. If so, the function that updates the game should stop running and text should be rendered on screen informing the user that they lost.	
FR_TELEPORT_PADS	The rooms in the game shall have teleport pads that the user can teleport between	UR_TELEPORT_PADS	There should be a code statement checking if the player's x and y position variables are inside the teleporter space, their x and y position variables should be changed to be within the field of the next teleporter.	

Non-Functional Requirements

ID	**Description**	**User Requirements**	**Fit Criteria**	**Priority**
NFR_DEVELOPMENT_TIME	The game must be completed by the deadline.	UR_DEVELOPMENT_TIME	The entire project, including the code and documentation must be submitted to the customer before the 25th of December.	Shall
NFR_2D	The sprite that the user controls should be able to move around in a simulated 2D space.	UR_2D	Sprites in libGDX have variables stating their X and Y positions on the screen. Motion on a 2D plane should be simulated by changing these values.	Shall
NFR_FIRST_TIME_USER	The game must be easy for a	UR_FIRST_TIME_USER	There must be a screen accessible from the main menu that tells the	Shall

	first-time user to learn.		user which controls to use.	