Software Requirements Specification

2.1 Introduction

Where possible we followed IEEE Recommended Practice for Software Requirements Specification [1] to ensure continuity, reduce ambiguity and have improve completeness of the requirements. The purpose of this document is to help us to form a plan for developing architecture and implementation, to aid testing and it will be used for checking against the system to test completeness. The specification will regularly be updated if/when requirements change, to help direct our development towards a solution that matches our client's mental model. The expected audience for this specification includes our stakeholders, academic staff at the University of York, and other students within our cohort. We have tried our best to word this document in a way sensible to our expected audience.

2.2 Requirement formation

The requirements for this project were initially extracted from the Scenario Brief [2], and then to define both functional and nonfunctional requirements in more detail, we constructed Use Cases [3]. These allowed us to generate scenarios to better help us to understand how users may use our system, and what we would have to do to successfully implement one that provide a positive user experience.

Where requirements in the brief seemed unclear we interviewed stakeholders to ask for clarification or elaboration [4]. In some cases where the requirements were clear but deemed unrealistic, we negotiated with stakeholders with the intention of delivering a more accurate and complete set of requirements. For example, the brief states that once a suspect has been wrongly accused, the user cannot interact with them for the rest of the game. We decided that the user may have a frustrating experience in certain circumstances due to this rule and it may become more difficult to complete the game. After an interview, our client agreed that this requirement was perhaps too complicated so we negotiated the details of the feature. We then created a prototype and used this to interview our stakeholders again. We compromised on how complicated the story should be and decided to tell the player when they have acquired the murder weapon.

The requirements are presented in a table so that the associated system requirements, environmental assumptions, risks and alternatives and a test description to each user requirement could be easily seen and tracked. This allows us to have a clear referencing system which will aid in the testing, review and evaluation stages of the game's development, while also being clear and concise in order to minimise confusion or misinterpretation.

2.3 Assumptions

To keep the requirements as simple as possible we have made the following assumptions: the user is a fluent English speaker, can operate a computer with a mouse and keyboard, and the system will be run on a Windows 10 PC with all relevant software installed (eg. Java Runtime Environment).

2.4 Considerations

There are many aspects of software engineering that are important to ensure the product appeals to a large market (for example accessibility, artwork, immersiveness, multiplayer etc.). However, as careful consideration and time is needed to research and develop each section, and this particular project focusses very little on these requirements, we elected to focus more on improving the standard of our core requirements.

2.5 Requirements Specification

**We have abbreviated Ron Cooke Hub to “RCH”;*

**We have abbreviated Non-Player Character to “NPC”;*

**We have abbreviated the SEPR 2016-17 Scenario 2 Brief [2] to “SB”.*

**All interview questions we refer to can be found on our website [4].*

No.	User Requirement	Associated System Requirements	Environmental Assumptions, Risks & Alternatives	Justification
1	Each playthrough should be different in some way to the user.	a. The system should be able to randomly generate new stories alternating murderers, motives, clue locations and methods.	Risk: There is a small chance that the user may play the same game again due to random error.	This feature is a requirement because without it, our game would have no replay value as the users would only have one story to play through. It is also explicitly written into the SB.
2	The player should be given options at the start of the game including Play, Leaderboard, settings etc.	a. The system should present a menu with buttons to begin the game, display the leaderboard and tutorials.	Assumption: The user has an input device to navigate the menus.	Having a menu will allow the users to choose when they would like to start their game & score timer. Our research revealed no games that did not have a main menu, so it should be a familiar concept with our users.
3	The user must be able to learn how to play the game from within the software.	a. The system should provide simple tutorials/tooltips to the user to help them to start playing the game quickly.	Assumption: The user gains enough understanding to play the game from the tutorial. Risk: The user does not understand the game after playing the tutorial.	All similar games we researched had built quick-start tutorials so the user should not have to resort to reading the user manual if they would like to quickly play the game.
4	The user must be able to customize the personality of their character.	a. The system should allow the selection of personalities	Assumption: The user has an input device to allow them to make the selections.	The SB describes this functionality, although the traits could be selected from a list of presets (i.e. choose which detective to play as), selecting traits to include/ exclude (i.e. using radio buttons), or by using user defined rankings (eg sliders)
5	The user must be able to navigate to all rooms in the RCH.	a. The game must include at least 8 rooms in the RCH. b. The system should present the user with a Graphical User Interface. c. The system should provide a way to navigate through different rooms of the RCH.	Assumption: The user has means of moving the mouse cursor and input.	The SB says we should “divide the Hub into at least eight rooms”, and based on game research, the player should travel through graphical representations of each of the rooms to discover clues.
6	The user should be able to find and collect clues within the Ron Cooke Hub.	a. The game should hide discoverable clues throughout the building based on the story. b. There must be at least one clue in each of the rooms in the RCH. c. Clues should be removed from the environment and saved in the player’s inventory once collected.	Risk: The user might struggle to find the clues. Alternative: The user has already found all clues.	The SB specifies this in detail. These clues are collected by the user and are the fundamental concept for winning the game. Research into murder mystery and detective stories revealed the “20 Rules For Writing Detective Stories” [5] which states that all clues must be “plainly stated and described” (rule 1)
7	The user should be able to review the evidence that they have already found	a. The system should provide a place where the clues the user has already found can be looked at and reviewed	Alternative: the user has no evidence to view, and the system should display a warning message.	The game is not one based on memory, thus there must be a way of viewing previously collected clues. The SB states that not all clues must point to the murderer, some may act as red herrings.
8	The user should be able to see and interact with other game characters and objects to help them in their quest for evidence.	a. The game must contain at least 10 NPCs, each with distinct personalities. b. The system should provide a way for the user to interact with NPCs/ objects, and they will respond dynamically. c.The game should hide clues that can be found by interacting with the NPCs.	Assumption: The character/ object is relevant to the story and isn’t part of the background.	The SB describes this interaction in depth. If the player could not interact with objects or NPCs they would have a very linear experience, so this interaction should help keep the players involved with the game.

9	The user should be able to question NPCs in different ways, and will receive varying amounts of help from them, depending on their question choice.	a. The game needs a system for conversations between the detective and NPCs. b. There needs to be a way of ending a conversation. c. The game should allow for different methods of questioning with the same NPC depending on the personality of the detective. d. The NPCs should react differently to one method of interrogation than another.	Risk: If they have accused the NPC, they will be unable to gain the required evidence to continue. Alternative: The NPC has already given the player all information.	The SB describes this action in depth too. Our research into other similarly styled games and their dialogue methods revealed most employ the selection of some sort of vague conversation topic to begin the interrogation, and subsequent replies are kept short. Often the selection screens offer replies in a different tense than what the detective says (eg. 'Accuse jane' is chosen, and the detective says 'I accuse you').
10	The user should be able accuse NPCs of being the murderer, with a motive clue and a weapon.	a. The game should provide a method of accusing suspect NPCs of being the murderer. b. The game should reward the player for correct accusations, and penalize for incorrect accusations. c. The game should alert the user when they have discovered the murder weapon.	Assumption: If the user is unable to question a particular NPC for whatever reason, the NPC should make this clear to the player.	The SB specifies that correctly accusing suspects is the way the player can win the game, so a system to control this is necessary for the game. An interview with the client revealed that when the weapon has been collected, the user should know, and they shouldn't be left to figure it out which clue was the weapon.
11	The user can accuse any suspect, but if they are wrong, or have not found enough evidence, they cannot question the player until new evidence is found.	a. The system must keep track of accusations, and stop players being questioned after being accused, but it should not be possible to lock the game so there is no possible completion.	Risk: The user might accuse all NPCs, stopping them from questioning them and rendering them unable to collect the required evidence to continue.	The SB states that 'If they haven't gathered enough evidence, or the accusation is false, the player cannot question that character again for the rest of the game.'. We along with the stakeholders have decided this is too harsh so changed it to this requirement.
12	The user should 'win' the game when they have successfully accused the correct suspect with correct motive clue and weapon.	a. The game should end when the correct suspect has been accused. b. The user must collect at least the motive clue and a number of other clues to successfully accuse.	Alternative: The game should end if there arises a situation from which the player cannot progress.	Like in the classic board game Cluedo, even if the suspect has been selected correctly, they player has not won without correctly deducing the other clues.
13	The player should receive a score based on their performance.	a. The system should provide a score based on many factors including missteps and time taken. b. The system should allow the user to enter their name. c. The game should save this score & name for viewing again in the leaderboard. d. They should be saved in a way that will preserve the data when the game is closed and started again.	Risk: The user leaves the game and forgets to pause. Penalty points keep getting added to their score, even though they are not present. Risk: If the user makes multiple mistakes or takes too long to complete the game, the counter could count too high. This could cause an overflow, which could crash the game if not handled correctly.	Other than being described in the SB, the score is an important aspect to any arcade-style game as it allows players to compete with others and themselves on successive attempts. Other games that we researched used scores so that
14	The player should be able to view previous scores.	a. Previously saved scores must be read by the system and form a simple leaderboard table.	Risk: There may not be any scores that have been previously saved. Assumption:If there are many with the same score then the list should be sorted in a reasonable order.	As above. If a user wants to try to beat their last score they'll need a device to store and recall this information; leaderboards are the most common way of representing this information since the invention of video games.
15	The player should be able to take a break without their score continually decreasing.	a. The game should provide a method of pausing the game	Risk: The player forgets to pause the game.	Every video game we researched featured some method of pausing the game to halt time and
16	It should be obvious to the player when the game is currently paused.	a. The game should display a new screen when it is paused b. There should be options to allow the user to resume the game, but also to exit to the menu.	Risk: The pause screen could look too similar to the play screen.	If it is not obvious when the game is paused, the player could become frustrated and confused as to why it is not working the way they expect.

2.6 Other System Requirements

No.	System Requirement	Environmental Assumptions, Risks & Alternatives	Justification
17	The system should clear the player's inventory when the game is restarted.	Assumption: The player's inventory isn't already empty to start with.	Different clues will be needed to accuse each game so having them remain would be unnecessary and their presence would compromise the story and clutter the inventory.
18	The system should randomly distribute the clues when the game is restarted.	Risk: Due to the nature of randomness, clues may appear in the same place they were previously found. Some clues should only be discoverable in a relevant location (eg. crime scene or specific room).	The clues should be in different places to make the game more challenging, and to make each playthrough unique.
19	The system should randomly select a murder weapon, and characters to be the murderer and victim when the game is restarted.	Risk: Due to the nature of randomness, these factors may not change from the previous playthrough.	Each factor should be randomised as stated in the SB. This allows each playthrough to be different, and varying stories can be generated by the engine, forming the dynamic content of the game.
20	The game must run on the computers found in the software lab.		One of the statements in the SB suggests that the University's Communications Office may use our software for Open Days and similar events. For this reason, our game should be compatible with University equipment, and should not be over demanding of processing/ graphics power.

2.7 Non-Functional Requirements

No.	Non-Functional Requirement	Justification	
21	The game should be interesting and fun to play.	SB states the game should be enjoyable.	
22	The game should not include any swearing, racism, sexism, ageism, or any other kind of discrimination or offensive language or images.	Risk: Cultural differences may mean that something innocent to one person is offensive to somebody with a different background.	Intentionally including any offensive material is morally wrong and could cause distress to our users. The SB also asserts that the University's Communications Office may use our software for UCAS and Open Days, and including anything offensive would result in bad publicity for the university, and an inaccurate portrayal of the Computer Science Department.
23	The game should be aimed towards university students	SB states the game should be playable by our SEPR cohort.	

2.8 Bibliography

[1] *"IEEE Recommended Practice for Software Requirements Specifications", Ieeexplore.ieee.org, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/720574/?part=1 . [Accessed: 20- Oct- 2016].*

[2] "SEPR 2016-17 Scenario 2", 2016. [Online]. Available:
https://henrycadogan.github.io/Clued-up/ass1/webfiles/SEPR2016-17Scenario2.pdf
[Accessed: 5- Oct- 2016].

[3] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Pearcy, "Use Case", Wedunnit!, 2016. [Online]. Available:
https://henrycadogan.github.io/Clued-up/ass1/webfiles/UseCase.pdf
[Accessed: 2- Nov- 2016].

[4] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Pearcy, "Interview", Wedunnit!, 2016. [Online]. Available:
https://henrycadogan.github.io/Clued-up/ass1/webfiles/Interview.pdf
[Accessed: 5- Nov- 2016].

[5] *Van Dine, S. S. "Twenty Rules For Writing Detective Stories". American Magazine September 1928. [Online]. Available:
http://www.dozenten.anglistik.phil.uni-erlangen.de/~cnhuck/Van%20Dine_Twenty%20rules%20for%20writing%20detective%20stories.pdf [Accessed: 20- Oct- 2016].*