University of York

Department of Computer Science

SEPR - Assessment 1

Requirements

Team Craig

Thomas Burroughs

Huw Christianson

Joseph Frankish

Isaac Lowe

Beatrix Vincze

Suleman Zaki

Requirement Elicitation Process

As specified in our Method Selection and Planning section, the team will be following an agile method, namely an adaption of the Scrum method. This method promotes face-to-face meetings and communication with the customer. On receiving the written brief, the team gathered to brain storm potential requirements that the game must fulfil and note questions about ambiguous statements in the brief that must be clarified on meeting with the customer. These requirements were often vague and required more elaboration. During the first meeting with the customer the noted questions were asked and answers regarding the clarification of requirements were noted so further discussion with the team could commence. From these answers, the team discussed potential design ideas for the game and how these ideas would meet the requirements of the customer. On research of requirements engineering it became clear that prototyping could help to elicit requirements, as well has give an insight into potential software architecture. Due to this each team member was asked to create and provide a paper prototype of a specific game feature. These paper prototypes were presented to the customer at the next meeting. Discussion and negotiation at the second meeting, prompted by the paper prototypes, elicited more requirements that the game should have, however, both the team and customer agreed that these requirements were optional.

Once all feedback from the customer had been discussed, the team created a Single Statement of Need (SSON) which was later agreed upon with the customer. This statement gives a comprehensive understanding of the desired outcome of the game/project and can provide a simple testable measure of how successful the requirements process have been completed. For this project the agreed SSON is as follows: “The system will deliver a game about zombies, given in a top-down perspective, with a novel mechanic.”

Requirement Specification and Presentation

A software requirements specification (SRS), is a detailed description of a software system to be developed with its functional and non-functional requirements. Our SRS was developed based off the agreement that was formed in the meetings between our team and the customer, as described in the elicitation process detailed above. Using the recommended IEEE practice for system requirements specification, our SRS will adhere to the following: correct; unambiguous; complete; consistent; verifiable; modifiable and traceable.[1] Our requirements are represented in a table to improve legibility and minimise documentation to adhere to our agile method. Each requirement within the table has an accompanied fit criterion. The fit criterion is used to quantify or measure the requirement which makes it testable, which will allow the team to determine whether a specific implementation actually meets the requirement [2]. Furthermore, alongside each requirement are environmental assumptions, risks and alternative for that specific requirement. This will help to assess and minimise potential risks that may arise. Alternatives provide evidence of further requirement elicitation and is indicative of the decision process. Alongside each given requirement within a table is a distinct requirement ID. This requirement ID allows the requirement to be traceable throughout all of the project’s documentation. Each requirement has been categorised as either ‘functional’ (F) or ‘non- functional’(NF), and further categorisation of the non-functional category has occurred with, for example, performance (P) , constraint(C) and usability (U) requirements. More non-functional requirement categories are likely to expand as more requirements are developed.

Requirement ID	Requirement Description	Fit Criterion	Environmental Assumptions, Risks or Alternatives
F1	Game must incorporate a novel mechanic	Game will contain at least one unique mechanic that differentiates this product to those produced in a similar project or products of a similar style (top-down shooter).	The project is to be used in between other, highly similar projects, necessitating a factor to make it stand out from others. As is, it should not be a higher priority than making a functional product that meets client requirements, but if the mechanic(s) are crucial to system use, then neglecting them at an early stage of the project can make it difficult to work with down the line
F2	Game must include a minigame, distinct from the main game.	Game contains a mini-game, of a different game style and design, can be entered by clicking on a graphic on the world map. The mini-game can be replayed as many times as the player wants.	Assume that like the main game, that the mini-game is not too graphically intensive and can be run of the pc's in the software lab.
F3	Game shall contain 5 powerups	The game will contain the following 5 power-ups: heal quicker; damage boost; speed up; nuke; insta-kill. These will can be acquired using virtual currency.	Trying to reach too far with the powerups risks wasted time making unnecessarily complicated. Risk that the powerups do not provide enough trait enhancements to make it worthwhile
F4	Game must have 6 six different locations.	The game will have 6 distinct locations based on real-life locations at the University of York. They are as follow: Ron Cooke Hub with Lake, Langwith, Goodricke, Constantine and the Retail Park.	Risk that some locations may be too similar to each other making gameplay seem repetitive.
F5	Game shall contain 3 different characters	The player can select from a panel of 3 characters all of which have a distinct visual appearance. Each character will have different traits such as greater damage, greater speed and faster healing.	Attempting to implement character traits that are too complicated can be risky as time could be wasted on coding them instead of simple traits. Character traits will be kept simple. Risk that characters' traits are not evenly matched or distinct enough leading to the players only using one player.
F6	Game must contain 2 bosses	The player will have to engage in combat with first a massive goose living below the Ron Cooke Hub Lake and then Koen	Risk that bosses too overpowered due to enhanced traits, meaning the bosses are

	that differ from regular enemies	Lamberts (with a goose's head) on entering the Computer Science building. Both bosses can be beaten but have enhanced traits. .	too difficult to beat causing the player to lose motivation. Risk that bosses will be too underpowered due to not enhancing traits enough, restricting player stimulation.
F7	Player must be able to win the game after visiting all 6 locations and defeating all bosses	The player can complete the game once the player has unlocked all location achievements by visiting each location and defeating Koen Lamberts and the giant goose.	Assume that all 6 locations are visitable and that both bosses are beatable. Assume that the player is aiming to complete the game. Risk that the player does not attempt to finish the game through to the end, possibly due to alternative investments in the game (i.e. mini-game) or the lack of interest.
F8	The game must contain regular enemies	Game contains zombies (all of the same graphic design and attributes) which will roam the world map and distribute from the edges of the map. Zombies will approach the player when they enter the players viewing frame. Players will then be able to attack zombies by clicking in the direction of the zombie using the selected weapon.	Assume that the machine will be powerful enough to support an arbitrary number of these enemies. Assume the player is of a suitable age to be exposed to 'violent' themes.
F9	Game must contain a way to navigate between locations	The game has distinct pathways between the 6 locations similar to those on the Hes East Campus which players cannot stray from. Players can move characters within locations using the w,a,s,d or arrow keys.	Assume the machine has supported periphery. Alternatives include modelling for different keys and controller support
F10	Player must be able to control the character during gameplay	The player will be in control of the player whilst the game is running using keyboard controls. Players will only not be in control of players when the game is paused, the player is on the menu screen or during cutscenes..	Assumption that player is not away from keyboard whilst game is running. Alternatively, player is in control using a different peripheral such as a controller.
C1	Project will be completed in its entirety by Wednesday 01/05/2019	Implementation and deliverables will be finished by 01/05/19. A Gantt chart has been created given specifics on time allocations and future scheduling.	Risk D1 and D2 (See Risk Assessment and Mitigation)
F11	Use of the lake location on campus will allow the player to access a fishing minigame	Game contains a graphic at the boundary of the Ron Cooke Hub lake which allows the player to enter the mini-game whilst pausing the main game.	Alternatively, this can just be accessible through menus either when the game starts or through the menu which appears when the game is paused. Risk that players cannot find minigame portal on the world map or do not recognise the

			portal as a link to mini-game feature.
C2	Game must run on Windows 10	Game runs on pc's in the software lab running on the Windows 10 operating system.	Assume players will choose to run Windows 10 on the Software lab pc's. Risk that pc's in the software lab may update or switch operating systems making the game potentially unplayable.
P1	Games must run smoothly on software lab computers	Game can run at a minimum of 30 FPS throughout the entirety of gameplay on the pc's in the Computer Science software labs.	Assume that the pc's in the hardware have at least the minimum hardware to run the game at 30 FPS and can use this as a benchmark for other pc's. Risk that changes to the hardware of the pc's in the software lab means the game cannot run at 30FPS or game cannot run at all.
F12	Game shall include virtual currency	The player can collect coins on route to the 6 locations and the two bosses and use these to trade for Items at the retail park.	Risk that tradable items do not provide a clear benefit for the player, resulting in a lack of interest in collecting virtual currency. Alternative could be for players to forage and search for food/health packs to improve their stamina and health- Risk that this could be too tedious.
U1	Game Menu must be intuitive and easy to navigate	New users shall be able to select a character and enter the world map on the first attempt at using the game within 1 minute.	Assume that the player is not impaired and capable of operating keyboard and mouse controls. Assume that the loading sequence of the game, including cutscenes, does not exceed 1 minute.

References

[1] IEEE, 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*. Institute of Electrical and Electronics Engineers, 1998.

[2]- S.Robertson et al. *Mastering the Requirements Process*. Sep 2012. Available at: http://www.informit.com/articles/article.aspx?p=1929849&seqNum=7