Requirements Document

Introduction (elicitation, negotiation, research)

In order to elicit our initial requirements, we took several steps in an attempt to understand what the customer wanted us to achieve. The first way in which we did this was targeting the assessment brief, which listed the basic needs of the system specified by the customer. From this, we gained many user requirements such as the using six characters and having a world map. Highlighting this gave us a basis in which to discuss these with the customer and negotiate further requirements.

Upon having a customer meeting, we refined the requirements elicited from the brief to more specific user requirements, and from these obtained the system requirements, by thinking about what our system would need to do to meet the user needs. This also help us to elicit functional requirements by brainstorming as a team to evaluate what our system would need to do to meet the requirements of the customer. We also drafted some non-functional requirements as constraints upon the functional requirements.

In order to develop some specific requirements, we used several examples of already successful games and presented our ideas to the customer. Some particularly influential games we considered were *South Park: The Stick of Truth*, using the idea of side-scrolling and 2.5 D graphics, and *Swords and Sandals 2*, to envision a turn based 2D combat system. From these we created a paper prototype of our combat system and played through this with the customer. This helped us to elicit requirements such as how the multiple characters could complement each other in the combat system, and how engaging with enemies to initiate a separate interface specifically for battles.

Overall, through team brainstorming, analysing the brief, and developing these ideas with the customer we developed a list of User, System Functional and non-functional requirements upon which to develop the system. Using use-cases and scenarios, as well as UML models, we were able to express the customer's needs as best as possible through the following requirements.

1. **User Requirements:**

Ref. Number	Requirement	Environmental assumptions and Associated Risks
1.1	At least 6 different playable character types	We assume these can be created based on one character class. A risk that could arise could be that designing six different classes may be time consuming.
1.2	A top-down world map displaying all character's progress	We assume it will contain a fantasy element based upon the real campus locations. A risk would be that it would be obscured too much to be recognisable.
1.3	A combat mode, with group battles	We assume it will be seamless and have a different perspective to normal gameplay. A risk could be that the characters are overpowered.
1.4	Market/town implementing a trade system with loot/gold	We assume characters can acquire currency. A risk could be finding ways to exploit the market to make money.
1.5	10 different world locations	We assume they will resemble university locations, a risk could be that they are not all reachable within short demonstration times
1.6	Ability to encounter enemies throughout the world	We assume they will be balanced by area and player level. A risk could be that the player may be able to avoid certain enemies and access further areas whilst too weak.
1.7	The ability to find treasure/loot	We assume the characters will be encounter random loot levelled to them. A risk could be there is too much/little loot available, or if loot refreshes too often.
1.8	Mini game relevant to the story	We assume the game will be relevant to the university of York theme. A risk could be it may be too complicated and take focus away from making the main game.
1.9	Load/Save the game at anytime	We assume this can be done anytime outside of combat. A risk could be if no autosave feature is implemented players may accidently loose progress.

2. System Requirements

Ref. Number	Requirement	Environmental assumptions and Associated Risks
2.1	The user should be able to customise a character to some degree	Assuming the character customisation will be quick enough to demonstrate a playthrough at open day. The risk is making the customisation too complicated.
2.2	The system will have a GUI that will display the chosen character + companions (using side-scrolling format)	We assume that the character will have its own art style and look depending on user choice.
2.3	Different character classes will represent the different roles of the characters	We assume users will be able to choose specific classes and work together with companions. A risk is that some classes may become over-powered.
2.4	The system will have a viewable map that can be accessed anytime, showing icons that represent the progress of the 6 playable characters	We assume all characters progress can be viewed on the map. A risk is that it may become congested with 6 playable characters all at the same time.
2.5	Interacting/choosing to attack an enemy initiates a separate combat mode. 3 columns for character + 2 companions, can move as grid, each character have own attacking abilities, enemies should drop loot + XP points	Assume characters have their own class abilities, and grid turn based combat system will be enjoyable for the user. A risk is that combat may take too long, or certain class setups will be overpowered.
2.6	Store within a town to trade loot for money and money for more items, ability to increase bartering to improve trade, items have an icon and a price	Assume we can create a balanced trade market, where prices are determined by bartering skill. A risk may be that player choices will skew prices too much (eg 0 bartering meaning lack of trade)
2.7	Display separate GUI for mini game, when interacted with in the world cuts to mini game, have ability to keep track of score, quit out at any time, gain reward from playing	Assume we will have time to implement appropriate minigame that can influence the characters skill. A risk could be that over-playing the mini-game may cause unfair advantage for some characters.
2.8	Have a pause menu where the progress of the game can be saved. Once game quit and reloaded, character appears where it was last saved with all same items	Assume that we can implement simple pause menu easy to save quickly. A risk could be no autosave features.

3. Functional Requirements

Ref. Number	Requirement	Environmental assumptions and Associated Risks
3.1	GUI displaying the character on screen	We assume it will adopt a 2.5D style, we feel this has little risk as it is a simplistic art style and can be done using unity.
3.2	Launch combat mode when player attacks an enemy/is attacked	Assume there will be optional and essential NPCs enemies. We risk making the combat mode too time consuming / must be relevant to the story.
3.3	Allow side-scrolling movement for the character (2.5D movement)	We assume we can do this once we draw up an art style (ideas influenced by *South Park: The Stick of Truth*). We risk not being original so need to create a unique art style.
3.4	Display a world map from the pause menu that displays progress (icons displaying character + companions)	We assume we can access the map at any time during the game and can build this in to the game through the pause menu. We risk the map being too complicated based on number of locations/characters.
3.5	Have the ability to trade items using loot/money system	Assume we can make a town in which players can trade items/some form of currency with NPCs. We risk this becoming exploited, need to make it balanced.
3.6	Character interaction – lead story getting quests/hints to lead to vice-chancellor and boss fights	We assume the character will play to the story line so events lead in to each other smoothly. We risk making the game too difficult without having certain hints or guides.
3.7	Experience point system allowing character to level up, use skill points to develop specific abilities for character types	We assume that the characters can gain skill points by completing quests and killing mobs, as well as bosses. We risk making the certain characters abilities better than others making some classes over-powered.
3.8	Ability to save progress at any time (not combat) and load back in at this point	We assume the player can save anytime but not combat as would be unfair and easy to exploit. We risk players not saving before big boss fights and losing large progress, may need autosave feature.
3.9	Display GUI for mini game that is implemented	We assume this will be relevant to story and provide character benefits. We risk this being exploited (getting too much XP for playing), or being too complicated (interrupting making of main game)

4. Non-Functional Requirements

Ref. Number	Requirement	Environmental assumptions and Associated Risks
4.1	Must be able to run on any operating system with a frame rate above x	We assume that through unity it will be available on all machines, and by play testing the game we can ensure good frame rates. A risk would the game being too slow, and unenjoyable for the user.
4.2	Movement should be immediate (quick response time), turn based combat	We assume that coding using unity will ensure we can make a smooth-running game. If the response time is slow we risk the game being unenjoyable and unable to demonstrate at open days for example.
4.3	The system must be reliable and not crash, losing player progress	Assuming we play test all areas of our game we should be able to eliminate any bugs causing this. The risk would be players losing all unsaved progress.
4.4	Shouldn't be an excessive file size (under x MB)	We assume that the code will not be too large in order to fit in most people's memory. The risk would a game to large would not be available for potential customers.
4.5	Must be delivered to the customer on 2/5/2018	We assume by using Agile methods we can stick to the deadline. The risk involved would be the customer receiving the product late.
4.6	Must be appropriate for students to play, not containing any offensive material	We assume that by taking basic precautions we can avoid any offensive material even though this is subjective. We risk offending members of the public at an open day if this requirement is not met.

5. Constraint Requirements

Ref. Number	Requirement	Environmental Assumptions and Associated Risks
5.1	Project – Financial (no money put in to the project, using resources available from the university), Project must be finished by 2/5/2018	We assume that all the tools we need to complete the project will be available or funded by the university removing financial investment/limits. We also assume that as a team we can deliver the completed project by the required date. If we do not have everything we need to complete the task we risk not delivering the product on time.
5.2	Design – Run on computers (any operating system), built using unity, based upon university campus, use keyboard, mouse or controller to play	We assume the way our system is built using unity allows it to be used on any operating system, and that it can be played using a controller if required. The risk is that if not the user may not feel they are having the most fun within the game, or risk not making it available to all potential buyers.

Use Case

Name: Engaging into combat

Context: A player approaching one or a group of mobs to trigger a battle within combat mode of the game.

Primary Actor: Player controlling the character

Precondition: Mobs are roaming around the world, and the player is in the area.

Trigger: Player attacks the enemy / Enemy attacks player

Main Success Scenario:

1. Player approaches enemy
2. Some action causes the player to engage in combat mode with chosen mob
3. Cuts to combat screen showing character and companions on one side, enemies on other side of the screen
4. Engage in turn based combat selecting different attacks/characters
5. Player wins combat and collects loot
6. Player progresses with game

Secondary scenarios:

2.1 Player cannot engage with enemy as in non-combat zone e.g. Market

5.1 The player loses the battle, and returns to the last saved location

Success Postcondition: Player successfully continues with the game

USE CASE DIAGRAM

Validation

Reviews, Prototyping