ENG1 Team Project
Assessment 2

Deliverable 2: Requirements

Team 13 – Team Unlucky

LILLY BROWN
ROSCOE GILLATT
ADAM JOHNSON
YUMIS ZYUTYU
BRANDON WEST
AYMAN ZAHIR

2. Requirements

Elicitation of requirements

1. The provided product brief indicated the overall goals and intentions of the finished product and contained general descriptions of its desired functionality
2. Group brainstorming session was held to compare our interpretations of the brief and to raise a list of questions to address to the customer
3. In a group meeting with the customer, answers to the questions and any other customer comments were detailed in informal meeting notes.
4. Recorded info was formalised as a set of user requirements.
5. User requirements were distilled down into more specific functional and non-functional requirements.
    a. Functional requirements detailed concrete, specific functionality and capabilities of the product as related to its software implementation.
    b. Non-functional requirements captured the performative characteristics of the completed product as a whole which could be perceived by the user or tester.
6. Resulting functional and non-functional requirements were closely evaluated for possible risks to their implementation; these were detailed within the risk register.

Research into requirement specification and presentation

- IEEE requirements engineering document[1]:
    - Provided comprehensive information on all aspects of requirement elicitation and presentation, although sections 5.1-5.2.8 and 6.1-6.6.3 were most helpful
    - Contained robust justification for the need for requirements and their role in the overall software lifecycle process
    - Informed our choice of specific language, standardised subjects and verbs (user, shall, etc.), the choice of imperative tone, and justifications for these
    - Ultimately, aimed at larger, more critical projects than ours
- ENG1 lecture on requirements engineering:
    - Provided an excellent overview of the motivations for requirements engineering and a lucid overview of this process
    - Introduced the user/functional/non-functional requirements methodology which proved an excellent fit for our scope of project (versus lifecycle-based requirement methodology in the IEEE document)
    - Demonstrated requirements tables as a tool for writing down and detailing requirements

Informed by these resources, we chose a tabular format for implementing for the requirements register, allowing us to easily add additional metadata to individual rows as extra columns and permitting a quick, comprehensive overview of the entire register. All rows were labelled with unique identifiers, permitting cross-referencing between user and (non-)functional requirements, as well as with other sections of project documentation, such as the risk register and architecture specs.

[1] *Systems and software engineering -- Life cycle processes -- Requirements engineering*, ISO/IEC/IEEE 29148:2018(E), 2018.

User Requirements

Category	ID	Description	Priority	Requirement Relations
Game setting	UR_PLATFORM	The user shall use a standard laptop PC to play the game	Shall	FR_VIEWPORT_SCALING / FR_MIN_FPS / FR_CROSS_PLATFORM_WIN / FR_CROSS_PLATFORM_MAC / FR_CROSS_PLATFORM_GNU_LINUX / NFR_COLOURBLINDNESS / NFR_NO_FLASHING
Game setting	UR_GAME_INIT	The user shall begin a new game from an initial state	Shall	FR_GAME_RESET
Game setting	UR_SHIP_CONTROL	The user shall control a ship sailing across the great Lake of York	Shall	FR_SHIP_KB_INPUT / NFR_USER_INPUT_LAG / NFR_RENDER_SMOOTHNESS
Game setting	UR_COMPETING_COLLEGES	The user shall encounter at least 3 other colleges	Shall	FR_COLLEGE_ENTITY_TRACKING / NFR_WORLD_COLLISIONS
Game setting	UR_LEARNING_CURVE	The user shall play the game without training	Shall	NFR_EASE_OF_USE
Game setting	UR_GAME_DURATION	The user shall be able to complete the game within a ~5 minute	Shall	NFR_GAME_DURATION
Game setting	UR_GAME_OBSERVABILITY	The game shall accomodate onlookers in the PCs surroundings	Shall	NFR_LARGE_ASSETS
Encounters	UR_FRIENDLY_SHIP_ENCOUNTER	The user shall encounter friendly NPC ships	Shall	FR_FRIENDLY_AI / FR_FRIENDLY_INTERACT
Encounters	UR_HOSTILE_SHIP_ENCOUNTER	The user shall encounter hostile NPC ships	Shall	FR_HOSTILE_AI / NFR_SHIP_COLLISIONS / NFR_AI_LAG
Encounters	UR_FIRE_WEAPONS	The user shall be able to fire weapons from the ship	Shall	FR_PLAYER_FIRE / FR_PLAYER_AMMO
Encounters	UR_BULLET_DODGE	The user shall be able to maneuver their ship to dodge fired munitions	Shall	FR_BULLET_TRAVEL / NFR_BULLET_COLLISIONS
Encounters	UR_FRIENDLY_BUILDING_INTERACT	The user shall interact with friendly buildings	Shall	FR_FRIENDLY_COLLEGE
Encounters	UR_HOSTILE_BUILDING_COMBAT	The user shall engage in combat with hostile buildings	Shall	FR_COMBAT
Encounters	UR_HOSTILE_COLLEGE_CAPTURE	The user shall capture other colleges via combat	Shall	FR_COLLEGE_CAPTURE
Earnables	UR_EARN_MONEY	The user shall earn money	Shall	FR_MONEY_TRACKING / FR_MONEY_UPDATE
Earnables	UR_EARN_POINTS	The user shall earn points	Shall	FR_POINTS_TRACKING / FR_POINTS_UPDATE
Progress	UR_QUEST_PROGRESS	The user shall progress through a series of quests	Shall	FR_QUEST_TRACKING / FR_QUEST_RANDOMISE / FR_QUEST_OBJECTIVE
Progress	UR_GAME_WIN	The user shall win the game through achieving an ultimate objective unlocked by the fulfilment of preceding requirements/quests	Shall	FR_BOSS_UNLOCK_TRACKING / FR_BOSS_SPAWN / FR_GAME_WIN
Progress	UR_GAME_LOSE	The user shall lose the game through being defeated in combat	Shall	FR_PLAYER_DEFEAT / FR_SCENARIO_FAIL
Encounters	UR_SHIP_COMBAT	The user shall engage in combat with other ships	Shall	FR_COMBAT
Encounters	UR_OBSTACLE_ENCOUNTER	The user shall encounter obstacles while sailing in game	Shall	FR_OBSTACLES
Encounters	UR_WEATHER_ENCOUNTER	The user shall encounter bad weather while sailing	Shall	FR_WEATHER_ENCOUNTER
Earnables	UR_SPEND_MONEY	The user shall spend the money earned	Shall	FR_SHOP
Earnables	UR_POWER_UPS	The user shall be able to find 5 power ups, including temporary immunity and repairing damage	Shall	FR_POWERUPS
Game setting	UR_GAME_DIFFICULTY	The user shall be able to select from 3 difficulties from game start (Easy, Normal, Hard)	Shall	FR_GAME_DIFFICULTY
Game setting	UR_SAVE_POINT	The user can save the game state	Shall	FR_SAVE_POINT

Functional Requirements

ID	Description	User requirement	Priority
FR_VIEWPORT_SCALING	The game shall render on a 13"-27" monitor	UR_PLATFORM	Shall
FR_MIN_FPS	The game shall render at a minimum of 30 FPS	UR_PLATFORM	Shall
FR_CROSS_PLATFORM_WIN	The game shall be playable on Windows	UR_PLATFORM	Shall
FR_CROSS_PLATFORM_MAC	The game shall be playable on Mac OS	UR_PLATFORM	May
FR_CROSS_PLATFORM_GNU_LINUX	The game shall be playable on GNU/Linux	UR_PLATFORM	Shall
FR_GAME_RESET	The game shall allow restarting play from an initial configuration	UR_GAME_INIT	Shall
FR_SHIP_KB_INPUT	The game shall accept keyboard input for ship control	UR_SHIP_CONTROL	Shall
FR_COLLEGE_ENTITY_TRACKING	The game shall keep track of ships and buildings for a minimum of 3 distinct factions	UR_COMPETING_COLLEGES	Shall
FR_FRIENDLY_AI	The game shall control the actions of friendly ships	UR_FRIENDLY_SHIP_ENCOUNTER	Shall
FR_FRIENDLY_INTERACT	The game shall allow user interaction with friendly ships	UR_FRIENDLY_SHIP_ENCOUNTER	Shall
FR_HOSTILE_AI	The game shall control the actions of enemy ships	UR_HOSTILE_SHIP_ENCOUNTER	Shall
FR_PLAYER_FIRE	The game shall enable the user to fire ship weapons	UR_FIRE_WEAPONS	Shall
FR_PLAYER_AMMO	The game shall maintain the state of the user's ship armament and ammunition	UR_FIRE_WEAPONS	Shall
FR_BULLET_TRAVEL	The game shall render the travel of a ship's fired munition	UR_BULLET_DODGE	Shall
FR_MONEY_TRACKING	The game shall keep track of a player's money	UR_EARN_MONEY	Shall
FR_MONEY_UPDATE	The game shall give money on success in quests and encounters	UR_EARN_MONEY	Shall
FR_POINTS_TRACKING	The game shall keep track of a player's points	UR_EARN_POINTS	May
FR_POINTS_UPDATE	The game shall give points with time survived and obstacles navigated	UR_EARN_POINTS	May
FR_QUEST_TRACKING	The game shall maintain the state of the user's progress through multiple objectives	UR_QUEST_PROGRESS	Shall
FR_QUEST_RANDOMISE	The game shall randomise user's objectives between different playthroughs	UR_QUEST_PROGRESS	May
FR_QUEST_OBJECTIVE	The game shall associate quest objectives with game entities	UR_QUEST_PROGRESS	Shall
FR_BOSS_UNLOCK_TRACKING	The game shall monitor quest progression status prior to unlocking final objective	UR_GAME_WIN	Shall
FR_BOSS_SPAWN	The game shall spawn boss upon final objective ready status	UR_GAME_WIN	May
FR_GAME_WIN	The game shall display game stats upon successful completion of boss encounter	UR_GAME_WIN	May
FR_PLAYER_DEFEAT	The game shall display game stats upon player defeat	UR_GAME_LOSE	May
FR_SCENARIO_FAIL	The game shall display game stats upon game over scenario completion	UR_GAME_LOSE	May
FR_GAME_DIFFICULTY	The game shall have three difficulty levels for the game and AI	UR_GAME_DIFFICULTY	Shall
FR_SAVE_POINT	The game shall be able to save the state of the game at any point so it can be reloaded	UR_SAVE_POINT	Shall
FR_SHOP	The game shall have a shop where the user can spend their money	UR_SPEND_MONEY	Shall
FR_POWERUPS	The game shall contain five powerups that the user can pick up	UR_POWER_UPS	Shall
FR_BAD_WEATHER	The game shall provide bad weather encounters to hinder the player	UR_BAD_WEATHER	Shall
FR_OBSTACLES	The game shall contain obstacles the player will need to navigate	UR_OBSTACLE ENCOUNTER	Shall
FR_COMBAT	The game shall allow for combat to occur between friendly and enemy ships and colleges	UR_COMBAT	Shall
FR_COLLEGE_CAPTURE	The game shall allow the user to capture an enemy college after defeating it	UR_HOSTILE_COLLEGE_CAPTURE	Shall
FR_FRIENDLY COLLEGE	The game shall allow the user to interact with friendly colleges	UR_FRIENDLY_BUILDING_INTERACT	Shall

Non-Functional Requirements

ID	Description	User requirement	Fit criteria
NFR_SHIP_COLLISIONS	The game shall detect collisions between different ships	UR_HOSTILE_SHIP_ENCOUNTER	Distance between drawn assets <5px
NFR_WORLD_COLLISIONS	The game shall detect collisions between ships and world objects	UR_COMPETING_COLLEGES	Distance between drawn assets <5px
NFR_BULLET_COLLISIONS	The game shall detect collisions between game entities and fired munitions	UR_BULLET_DODGE	Distance between drawn assets <5px
NFR_USER_INPUT_LAG	The game shall be responsive to user input	UR_SHIP_CONTROL	Input lag <200ms
NFR_AI_LAG	NPC actions' responsiveness shall approximate that of player actions	UR_HOSTILE_SHIP_ENCOUNTER	AI response time <200ms
NFR_RENDER_SMOOTHNESS	The game world shall render smoothly during player movement	UR_SHIP_CONTROL	Visual render lag <200ms
NFR_COLOURBLINDNESS	Game map and assets should be distinguishable by a colourblind person	UR_PLATFORM	Subjective screenshot test via colourblind accessibility evaluation app
NFR_EASE_OF_USE	The game shall be self-explainable and feature obvious controls	UR_LEARNING_CURVE	Tester must be able to pick up and play with no prior instruction
NFR_GAME_DURATION	The game shall finish within ~5 mins in a win or loss for the player	UR_GAME_DURATION	Tester must reach the game stats screen within 4-6 mins
NFR_LARGE_ASSETS	The game assets shall be large enough to observe	UR_GAME_OBSERVABILITY	Observer standing 2m away
NFR_NO_FLASHING	The weather effects in the game should not flash, as it could cause epilepsy	UR_PLATFORM	Observer should not see any flashing