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.
    i. Functional requirements detailed concrete, specific functionality and capabilities of the product as related to its software implementation.
    ii. 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 labeled 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.

ID	Description	Priority
UR_PLATFORM	The user shall use a standard laptop PC to play the game	Shall
UR_GAME_INIT	The user shall begin a new game from an initial state	Shall
UR_SHIP_CONTROL	The user shall control a ship sailing across the great Lake of York	Shall
UR_COMPETING_COLLEGES	The user shall encounter at least 3 other colleges	Shall
UR_LEARNING_CURVE	The user shall play the game without training	Shall
UR_GAME_DURATION	The user shall be able to complete the game within a ~5 minute timespan	Shall
UR_GAME_OBSERVABILITY	The game shall accomodate onlookers in the PCs surroundings	Shall
UR_FRIENDLY_SHIP_ENCOUNTER	The user shall encounter friendly NPC ships	Shall
UR_HOSTILE_SHIP_ENCOUNTER	The user shall encounter hostile NPC ships	Shall
UR_FIRE_WEAPONS	The user shall be able to fire weapons from the ship	Shall
UR_BULLET_DODGE	The user shall be able to maneuver their ship to dodge fired munitions	Shall
UR_FRIENDLY_BUILDING_INTERACT	The user shall interact with friendly buildings	Shall
UR_HOSTILE_BUILDING_COMBAT	The user shall engage in combat with hostile buildings	Shall
UR_HOSTILE_COLLEGE_CAPTURE	The user shall capture other colleges via combat	Shall
UR_EARN_MONEY	The user shall earn money	Shall
UR_EARN_POINTS	The user shall earn points	Shall
UR_EARN_XP	The user shall earn XP	May
UR_QUEST_PROGRESS	The user shall progress through a series of quests	Shall

ID	Description	Priority
UR_GAME_WIN	The user shall win the game through achieving an ultimate objective unlocked by the fulfilment of preceding requirements/quests	Shall
UR_GAME_LOSE	The user shall lose the game through being defeated in combat	Shall
UR_SHIP_COMBAT	The user should be able to engage in combat with other ships	
UR_OBSTACLE_ENCOUNTER	The user may encounter obstacles while sailing in game	
UR_WEATHER_ENCOUNTER	The user may encounter bad weather while sailing	
UR_SPEND_MONEY	The user should be able spend the money earned	
UR_POWER_UP	The user should be able to obtain power ups through either the shop or at random locations on the map.	
UR_DFCLTY_LVL	The user should be able to choose from 3 difficulty levels (e.g. easy, normal, hard)	
UR_GAME_SAVE	The user should be able to save the state of the game at any time and be able to resume it at a later point.	

```
**Functional Requirements**
```

Description	User requirement	Fit criteria	Risks
The game shall detect collisions between different ships	UR_HOSTILE_SHIP_ENCOUNTER	Distance between drawn assets <5px	R4
The game shall detect collisions between ships and world objects	UR_COMPETING_COLLEGES	Distance between drawn assets <5px	
The game shall detect collisions between game entities and fired munitions	UR_BULLET_DODGE	Distance between drawn assets <5px	

Description	User requirement	Fit criteria	Risks
The game shall be responsive to user input	UR_SHIP_CONTROL	Input lag <200ms	
NPC actions' responsiveness shall approximate that of player actions	UR_HOSTILE_SHIP_ENCOUNTER	AI response time <200ms	R3
The game world shall render smoothly during player movement	UR_SHIP_CONTROL	Visual render lag <200ms	R8
Game map and assets should be distinguishable by a colourblind person	UR_PLATFORM	Subjective screenshot test via colourblind accessibility evaluation app	
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	
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	
The game assets shall be large enough to observe from several metre's distance away on a standard laptop PC screen	UR_GAME_OBSERVABILITY	Observer standing 2m away should be able to answer questions about gameplay state	

```
**Non-Functional Requirements**
```

Description	User requirement	Fit criteria	Risks

Description	User requirement	Fit criteria	Risks
The game shall detect collisions between different ships	UR_HOSTILE_SHIP_ENCOUNTER	Distance between drawn assets <5px	R4
The game shall detect collisions between ships and world objects	UR_COMPETING_COLLEGES	Distance between drawn assets <5px	
The game shall detect collisions between game entities and fired munitions	UR_BULLET_DODGE	Distance between drawn assets <5px	
The game shall be responsive to user input	UR_SHIP_CONTROL	Input lag <200ms	
NPC actions' responsiveness shall approximate that of player actions	UR_HOSTILE_SHIP_ENCOUNTER	AI response time <200ms	R3
The game world shall render smoothly during player movement	UR_SHIP_CONTROL	Visual render lag <200ms	R8
Game map and assets should be distinguishable by a colourblind person	UR_PLATFORM	Subjective screenshot test via colourblind accessibility evaluation app	
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	
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	