UNIVERSITY of TASMANIA

Assignment 3 Agile Cycle 1 Report
Fast Rescue Boat (FRB) Partial Mission Simulator

KIT301 ICT Project A

FAST RESCUE BOAT (FRB) OPERATOR'S COURSE (AMCS, 2024)

Zac Partridge (597945), Sebastian Brain (576267), Iurii Terentev (622347) and Winston Stuart (496314)

Semester 1, 2024

Execute Summary

This report summarises the progress and findings of Agile Cycle 1 for the Fast Rescue Boat (FRB) Partial Mission Simulator project. The project aims to deliver a high-fidelity training simulator for FRB operators that simulates various environmental conditions. During Agile Cycle 1, significant progress was made in developing the initial prototype, focusing on user interface design, core simulation functionality, and preliminary testing. The report highlights successful outcomes in prototype development, positive client feedback, and areas for improvement to be addressed in subsequent cycles.

Table of Contents

Introduction

Project Background

The Fast Rescue Boat (FRB) Partial Mission Simulator is designed to improve the training of FRB operators by providing a realistic simulation of different weather and water conditions. This project addresses the limitations of current AMC Beauty Point campus training methods, aiming to enhance skill retention and preparedness. The simulator serves diverse users, including trainees, certified operators, AMC staff, and technical support personnel.

Purpose of the Report.

This report aims to detail the activities and achievements of Agile Cycle 1, providing insights into the design and development process. The report evaluates the project's progress, documents client feedback, and outlines necessary changes for future development cycles.

Structure of the Report

The report is structured as follows:

- **System Interfaces:** Analysis of design decisions and progress in interface development.
- **Functionality:** Discussion of architectural decisions and technical development progress.
- **Technical Testing & Quality Control:** Adherence to testing methodology, outcomes, and quality and compliance standards.
- **Client Feedback**: Process of gathering and analysing client feedback.
- **Review of Progress and Required Changes:** Evaluation of alignment with the project schedule, requirements changes, and project plan updates.
- **Conclusions:** Recap of critical findings and future directions.

System Interfaces

Interface Design Decisions

Best Practice Principles:

When designing the user interface, we followed industry best practices to ensure functionality and user satisfaction. These principles included:

- **User-Centric Design:** We prioritised the user experience by creating an interface that is intuitive and accessible for all users, regardless of their familiarity with similar software.
- **Consistency:** We maintained a consistent look and feel across all interface elements, such as buttons, fonts, and layout structures, to promote usability and prevent user confusion.
- **Simplicity:** The interface was designed to keep interactions simple and efficient, ensuring that users can achieve their goals in the fewest possible steps without feeling overwhelmed.
- **Visual Hierarchy:** We arranged everything on the screen to show what's most important first, using size and colour to make it easy to see where to focus. This helps users find what they need quickly and makes the interface friendly and straightforward.

Examples and Explanations:

We made several key design decisions in the interface that specifically cater to the needs of AMC and its simulator users.

- **Colour:** AMC uses a specific colour palette across its internal projects. We incorporated this palette into the interface design. We selected shades that complement the existing AMC palette for elements requiring additional colours to maintain visual harmony.
- **Navigation:** We designed the main menu with larger, bold text at the top of the screen, clearly guiding users. Submenus and secondary options appear in smaller sizes, making it easy for users to find what they need quickly and navigate the system intuitively.
- **Style and Realism:** The simulator focuses on a realistic representation of maritime environments. We adopted a grounded art style and incorporated high-quality, real-life images for mission objectives and sea conditions. This approach ensures that the interface aligns with the realistic nature of the simulator.
- **Branding:** To strengthen brand recognition the AMC business and game logos are prominently displayed on the title splash page to strengthen brand recognition. This is crucial for presentations outside the organisation, as it immediately identifies the product's affiliation with AMC.

- **Language:** The terminology used within the interface adheres to seafaring standards, enhancing the simulator’s authenticity. For example, the sea states selectable by users are listed as Calm, Moderate, and Rough, reflecting the typical descriptors used in maritime contexts. This familiar ordering helps users intuitively understand and interact with the simulator settings.

Technical Development Progress:

Progress in the technical development of these interfaces has been substantial:

- We have completed the initial implementation of the user interface, including all primary menu navigation elements and interaction frameworks consistent with the design principles outlined.
- AMC's logos are prominently displayed on the title splash page, ensuring immediate brand recognition upon launching the simulator. The placement enhances visibility while maintaining a professional look.
- Integration of realistic images and adherence to the AMC colour palette has been achieved, with adjustments made based on initial user feedback emphasising clarity and accessibility.
- We have completed the mission selector, which is now fully functional and allows users to choose their objectives quickly. Additionally, the sea state selection framework has been finalised, setting the stage for its upcoming integration.
- We've implemented a speedometer reading in knots and a digital compass in the interface. The speedometer updates in real time, offering clear visibility for monitoring speed, while the compass provides directional guidance. These tools integrate seamlessly with the overall design for a realistic and functional experience.

This structured approach ensures that the interface development meets and exceeds the expectations and needs of AMC and its simulator users, paving the way for successful project completion.

Functionality

Architectural Decisions

Best Practice Principles:

It was identified as an iterative and incremental process in the early stages of development. There were too many things to take care of in one agile cycle, and the things that would fit into the agile cycle would likely need to be changed upon feedback. The information given to us also needs to be completed or interpreted in different ways.
To account for the project's iterative and incremental nature, the team decided that the most important principles the design should uphold are flexibility and modularity. The project needed to be flexible to allow for quick and drastic changes based on feedback and new information, and it needed to be modular to facilitate this flexibility. Parts of the project should not be interdependent so that they can be easily changed or swapped out entirely. All of the design decisions during development were based on these principles.

Architectural Decisions:

General decisions:

For the first agile cycle, three systems were implemented. Those systems would have basic functionality in the first agile cycle and will be iterated upon as needed in the coming cycles based on the feedback received. The systems will **not** be connected and built separately to support modularity and flexibility. The systems are ocean and floating, movement of the boat, and mission system.

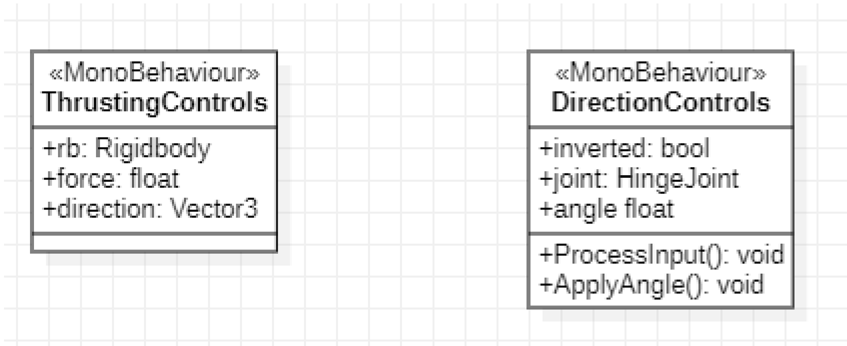
Ocean logic and floating physics:

Implementing ocean logic and floating physics would be a pervasive task. The client decided to use existing libraries to implement those in this project. The team used official documentation for the library to set up the ocean environment and the water objects in the scene. Values for buoyancy, mass, and mass distribution should be taken from the sources provided by the client.

Movement:

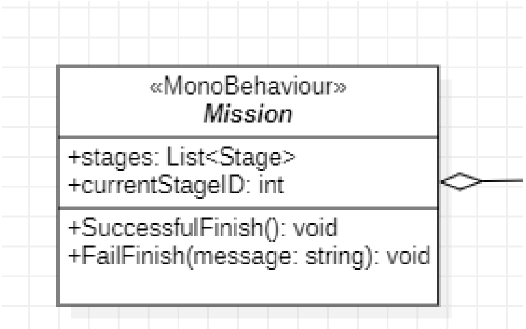
While implementing the movement system, the focus was on two things. First, the system needed to work for boats with jet and outboard propulsion, as the client highlighted that adding multiple boats could benefit the project. Second, it needed to be easy to test with a keyboard or a controller and connect to the controls given by

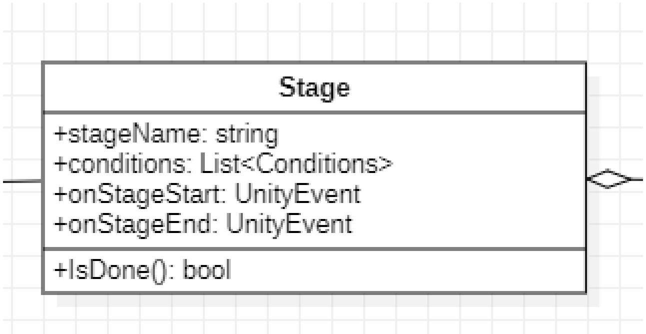
the client. This way, much time can be spent testing the movement system without waiting for the client to provide the controls.
The final design was quite simple, as can be seen below. It was decided to split the system into two separate behaviours that can be attached to different objects. The developers can change the direction in which the force will be applied, the force that should be applied (in Newtons), and if the direction controls should be inverted for the type of boat it will be used on.

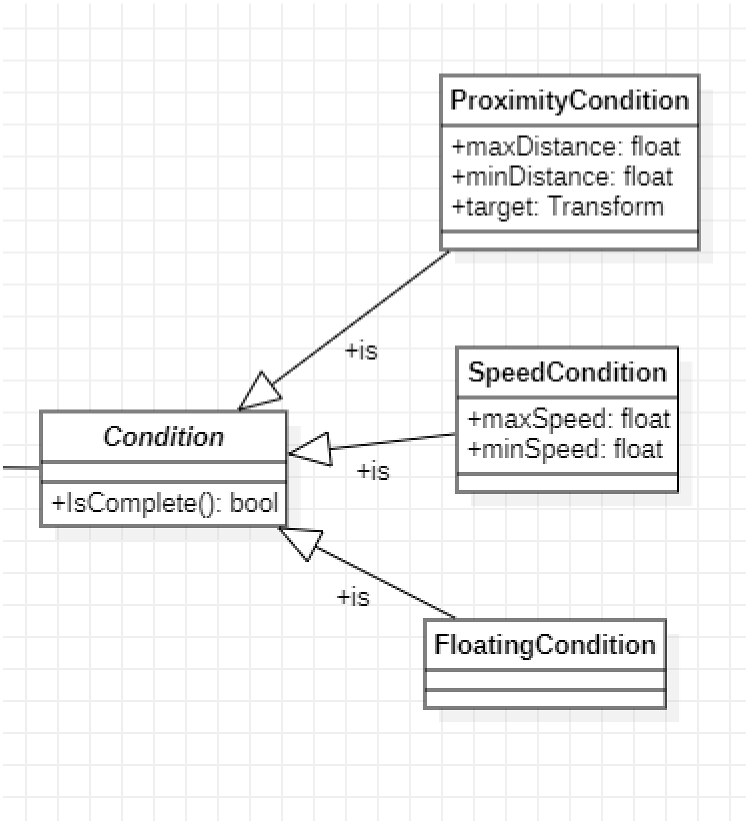


Mission system:

The mission system was the most extensive system developed. The mission system relies on three main classes: mission, stage, and condition. The mission class is in charge of storing all of a mission's stages and moving from one stage to the next when appropriate. It is also responsible for the logic of mission success or failure. Stages are appropriate for describing the stage, storing what conditions must be met to be considered finished, and triggering events when the stage starts and finishes. The class is not responsible for writing what should happen when the stage starts or finishes. Instead, the developer should create scripts with logic triggered by the events to support the modularity of the project. The condition class is abstract, and every class that extends from it must be able to tell whether or not the condition is met. This is done so the developers can easily and quickly create new small condition classes that are easy to test and implement into an existing system. The class diagram can be viewed below.







Technical Development Progress:

The project's technical development is on schedule. All systems have been implemented as planned and are fully functional.
The boat currently used for movement is jet-propelled. Both the direction and thrusting controls are attached to the nozzle. The controls work with keyboards and game console controllers. There

has yet to be an opportunity to test it with controls provided by the client. The thrust force is set to match the force from the engine documentation. The ocean environment has also been successfully set up to match the available documentation. Both of those could be reconfigured to match the feeling of handling a boat more accurately. There is currently only one mission implemented. The mission consists of the boat lowering into the water from a larger vessel, traversing it to pick up an object that simulates a rescuee, and returning to the larger vessel. The mission stages are boat lowering, rescue, and return to the boat. The conditions in use are collider, timer, and velocity. The mission works smoothly through all stages and is easy to modify and expand upon.

Technical Testing & Quality Control

Testing Methodology and Outcomes

Best Practice Principles:

To ensure the testing process was effective and thorough, we adhered to the following best practice testing principles:

- **Systematic Approach:** Implementing a structured and repeatable testing process to ensure comprehensive coverage of all functionalities.
- **Test-Driven Development (TDD):** Writing tests before developing the corresponding functionalities to ensure that the code meets the required specifications from the outset.
- **Continuous Integration (CI):** Integrating code changes frequently and running tests with each integration to detect issues early.

Testing Methodology:

- **Unit Testing:** Individual simulator components were tested in isolation to verify their functionality.
- **Integration Testing:** Combined components were tested to ensure they work together as expected.
- **System Testing:** The complete system was tested in an environment that mimics the production environment to validate the overall functionality and performance.
- **Regression Testing:** Ensuring new code changes do not adversely affect existing functionalities.

Testing Outcomes:

- **Menu Interface:** While the Condition Switching functions in the main menu correctly toggle between each state, no values are passed into the simulator. Mission Switching functions as expected, though mission “Pacing Exercise” is currently not implemented.
- **Simulator Physics:** While piloting the vessel at low speeds, it behaves as it would in an equivalent real-world setting. However, while piloting the vessel at or near its maximum speed, it begins to behave erratically, rolling over or becoming airborne.
- **Simulator Controls:** The vessel is correctly able to be controlled with both keyboard input as well as with a Xinput device, such as a controller or racing wheel, with throttle control bound to the analogue Y-Axis (up/down), and directional control bound to the analogue X-Axis (left/right)
- **Simulator Interface:** The vessel's current velocity is correctly displayed on-screen (converted to knots). While stationary, the heading is correctly displayed as a compass wheel. However, while moving, the compass wheel behaves erratically.

Quality and Compliance:

- Winston Stuart and Zac Partridge have primarily led the Quality Control process during Agile Cycle 1. They created a template for organising test criteria, documenting results, and ensuring all tested components follow appropriate software design principles.

Client Feedback

Feedback Elicitation Methodology and Outcomes

Best Practice Principles:

To elicit meaningful and actionable feedback from the client, we followed these best practice principles:

- **Clear Communication:** Ensuring all stakeholders were well-informed about the feedback process and its significance.
- **Structured Feedback Sessions:** Conducted organised sessions that allowed for comprehensive feedback collection.
- **Active Listening**: Engaging with clients to fully understand their perspectives and concerns.

- **Feedback Categorisation**: Organising feedback into specific categories to facilitate effective response and action planning.

Methodology

- **Initial Feedback Session**: The project team conducted a formal feedback session with the client, showcasing the progress made during Agile Cycle 1. This included a detailed presentation of the user interface design, core simulation functionality, and preliminary testing outcomes.
- **Follow-up Discussions:** Additional discussions were held to gather more detailed feedback and clarify any points raised during the initial session. These discussions took place both in person and through email exchanges.
- **Showcase Event:** An upcoming showcase event has been scheduled, inviting additional stakeholders to review the project. This event will provide an opportunity for real-time verbal feedback and further validation of the project's direction.
- **Review and Analysis**: The project team reviewed and analysed all feedback, identifying common themes and specific suggestions for improvement.

Feedback Outcomes

- **Positive Reception:** The client expressed high satisfaction with the project's current status, particularly praising the user interface design and core simulation functionality.
- **Clear Presentation:** The clarity and thoroughness of the project presentation were appreciated, with the client acknowledging the detailed walkthrough of the prototype.
- **Engagement and Support:** John's email inviting additional stakeholders to the project's first showcase reflected the client's support and enthusiasm. The scheduled showcase aims to engage ten more attendees, providing an opportunity for broader feedback and further validation of the project's direction.
- **Areas for Improvement**: The feedback included constructive criticism on specific areas. For instance, the client suggested that the user interface design could be tweaked to better align with other projects they are familiar with, enhancing consistency and ease of use. Additionally, they recommended enhancing the realism of environmental conditions and improving the overall user experience. These suggestions have been documented for incorporation into future development cycles.

Conclusion

Agile Cycle 1 of the Fast Rescue Boat (FRB) Partial Mission Simulator project has significantly progressed in several key areas. The project successfully advanced the development of system interfaces, core functionalities, and technical testing procedures, adhering to best practice principles. The user interface design incorporated industry standards for user-centric design, consistency, simplicity, visual hierarchy, and branding, ensuring users' functional and visually appealing experience.

The architectural decisions were guided by flexibility and modularity, resulting in three independent and fully functional systems: ocean logic and floating physics, boat movement, and the mission system. Based on feedback, each system was developed to support easy iteration and rapid adaptation. The technical development is on schedule, with the implemented systems performing as intended.

Testing methodologies, including unit testing, integration testing, system testing, and regression testing, ensured thorough validation of the simulator's components and overall performance. Despite some challenges, such as erratic vessel behaviour at high speeds and minor interface issues, the testing outcomes have provided valuable insights for future improvements.

Client feedback has been overwhelmingly positive, particularly regarding the user interface design and core simulation functionality. The structured feedback sessions and the upcoming showcase event have facilitated comprehensive feedback collection, highlighting areas for further enhancement, such as aligning the user interface with other familiar projects and increasing the realism of environmental conditions.

Moving forward, the project team will incorporate the feedback received into the next development cycle, refining the user interface, improving environmental realism, and addressing any remaining technical issues. Continually engaging with clients and stakeholders will ensure the project remains aligned with user needs and expectations, ultimately leading to an efficient and realistic training simulator for FRB operators.

References

-AMC Search, (2024), 'FAST RESCUE BOAT (FRB) OPERATOR'S COURSE', <https://www.amcsearch.com.au/course/fast-rescue-craft>, viewed on 12/03/2024.

-BasuMallick C, (2022), 'What Is TDD (Test Driven Development)? Process, Importance, and Limitations', <https://www.spiceworks.com/tech/devops/articles/what-is-tdd/>, Viewed on 7/05/2024

-CloudBees, (2023), 'What is Continuous Integration?', <https://www.cloudbees.com/continuous-delivery/continuous-integration>, Viewed on 7/05/2024

Appendices

Appendix A: Internal Testing Criteria, Methods and Results

QA Testing Criteria and Results - 22/5/24

Menu Function – Mission Switching

Criteria	Ensure that each mission option in the menu is
Method	Run both missions within the simulator using menu settings to ensure the mission loads correctly and can be exited appropriately using the 'Esc' key.
Expected Result	When selecting each mission, the user should be loaded into a specific scene containing that mission and able to exit the scene using the 'Esc' key quickly.
Result	When selecting each mission, the user is either loaded into the "Man Overboard" scenario (when the "Man Overboard" mission is selected) or into a legacy "Control Test" scene (If "Pacing Exercise" is selected). The user can exit from both scenes using the 'Esc' key.
Outcome	PASS*

* Will require scene "Control Test" to be replaced in Agile Cycle 2

Menu Function – Condition Switching

Criteria	Ensure that the ocean condition selected within the menu is accurately reflected within the Simulator
Method	For each mission, cycle through every condition setting to ensure that values are correctly passed to the simulator and that the simulator behaves appropriately to the setting.
Expected Result	Menu Elements cycle through correctly and pass correct values through to the selected simulator scene, which correctly applies values to the weather system to produce desired conditions
Result	Menu Elements cycle through correctly, but the function to pass values through to the simulator is not currently implemented.
Outcome	FAIL* - Function not Implemented

*UI Elements otherwise function as expected

Simulator Function – Vessel Controls

Criteria	Ensure that the vessel can be controlled using both Xinput devices as well as using keyboard input (if necessary)
Method	Using the XInput Device (Controller, Racing Wheel, etc.), Ensure that the Vessel responds to each input with varying degrees of power/steering, dependent on the User Input
Expected Result	Vessel thrust power should change by changes in Xinput Y-Axis (Left Analogue Stick Y-Axis), and Vessel thrust direction should change by Xinput X-Axis (Left Analogue Stick X-Axis)
Result	Vessel thrust power and direction change as expected from user input.
Outcome	PASS

Simulator Function – World Bounds

Criteria	Ensure that the user is unable to stray too far from the mission area
Method	Using either Keyboard or XInput controls, maneuver the vessel to each of the four cardinal edges of the map in any order.
Expected Result	Upon nearing the edge of the map, the user should be prompted to turn back via an on-screen message (i.e. “Leaving Mission Area”). Should the user attempt to proceed, the user should be teleported back to the mission area.
Result	The function currently not implemented - Unable to test
Outcome	FAIL - Cannot be Tested

Simulator Function – Vessel Physics

Criteria	Ensure that the vessel behaves as realistically as possible within simulated conditions
Method	Using the Keyboard or X-input controls, maneuver the vessel at varying speeds, combining gentle and harsh turns.
Expected Result	Vessels should behave realistically when accelerating and maneuvering.
Result	Vessel behaves realistically at low speeds but behaves erratically at high speeds, often rolling over or becoming airborne
Outcome	FAIL - Vessel does not behave correctly at high speeds

Simulator Function – UI Elements

Criteria	Ensure that Simulator UI Elements (Speedometer and Compass) display the correct values, according to Vessel velocity and heading, respectively.
Method	Using any input format, maneuver the vessel as usual. Observe UI elements and Debug messages to ensure that values match.
Expected Result	Vessel velocity (measured in kts) should correctly be displayed in the Speedometer (Lower Right), and the Compass (Lower Left) should correctly rotate to display the heading
Result	Vessel Velocity is correctly displayed in the Speedometer UI element, but the Compass UI element only correctly displays the heading when stationary; otherwise, it rotates erratically
Outcome	FAIL - The Compass UI element does not correctly function