NTNU
Kunnskap for en bedre verden

Department of Computer Science

Master Thesis

Developing a video game for research and prototyping of unmanned maritime vessels

Author:
Mikael R. Hanssen

Supervisor:
Yngve Dahl

Co-supervisors:
Ole A. Alsos
Erik A. Veitch

June, 2022

Preface

Acknowledgements

This project has not been a lone endeavour, and is the culmination of multiple research projects and would not be possible without a number of talented collaborators. I would like to express my gratitude to:

My supervisors, for continued guidance and support.

The researchers at SCL, for the opportunity to be part of the exciting development of autonomous ferries in Trondheim, and for continued access to their lab during development.

Erik Veitch, for his continued involvement and enthusiasm throughout the development process.

Sondre Ek, for a great collaboration during the preparatory project, and his contributions during development.

My family, for their great support and encouragement.

Abstract

Following recent development within autonomy and artificial intelligence, there is a considerable advancement within research for autonomous vehicles, especially for passenger transport. While self-driving cars have gotten major public attention, there is also noticeable development within the maritime domain.
While autonomous ships no longer require an onboard crew, there is still a need for a human to monitor and take control if necessary. Simulators are a great tool for evaluating the safety of such system, as extensive testing may be either too costly or too risky to be performed in real life.

This thesis focuses on the use of simulation and video game technologies for research and prototyping of unmanned maritime vessels. A simulator was developed, targeting an ongoing research project focusing on autonomous passenger ferries. The simulator was used for testing remote monitoring of autonomous ferries, using specific (simulated) scenarios designed to test extreme cases where the test participants was forced to take control of the ferry.
Malone's heuristics for designing instructional video games was used as a framework to evaluate the simulator.

Development has shown that using video game technology has accelerated the prototyping process, and helped to collect relevant data for further research. The analysis shows that using game design principles help with immersion a participant onto the role as a remote operator, and contributes to increased fidelity. However, one has to be cautious, so that the focus on games isn't detrimental to the accuracy of a simulator created for research purposes.

Sammendrag på norsk er på neste side

Sammendrag

I følge av nylig utvikling innen autonomy og kunstig intelligens, har det vært en betydelig framgang innen forsking for autonome kjøretøy, særlig innen passasjertransport. Selv om det er selvkjørende biler som har fått mest offentlig oppmerksomhet, er det også nevneverdig utvikling innen den maritime sektor.

Denne avhandlingen fokuserer på bruken av simulasjon og videospillteknologi for forskning og prototyping av ubemannede maritime fartøy. En simulator ble utviklet, med fokus på et pågående forskningprosjekt innen autonome passasjerferger. Simulatoren ble brukt for testing av fjernovervåking av autonome ferger, ved bruk av spesifikke (simulerte) scenarier som var designet for å teste ekstreme situasjoner hvor testdeltakerne ble tvunget til å ta over kontroll av fergen.
Malone's heuristikk for design av instruktive videospill ble brukt som et rammeverk for å evaluere simulatoren.

Utviklingen viste at videospillteknologi har akselerert prototyping, og hjalp til med å samle inn relevante data for videre forskning. Analysen viser at bruken av spilldesignprinsipper hjalp med å øke deltakernes innlevelse som rollen av en fjernoperatør, and bidrar til økt "fidelity". Derimot må man være varsom, slik at fokuset på spillaspektet ikke negativt går utover hvor realistisk og nøyaktig simulatoren er som et forskningsverktøy.

Simulator showcase

The simulator developed throughout this project has been used for various showcases, such as at UKATech, part of UKA 2021 in Trondheim (figure 1), and at NTNUs Shore Control Lab (figure 2).

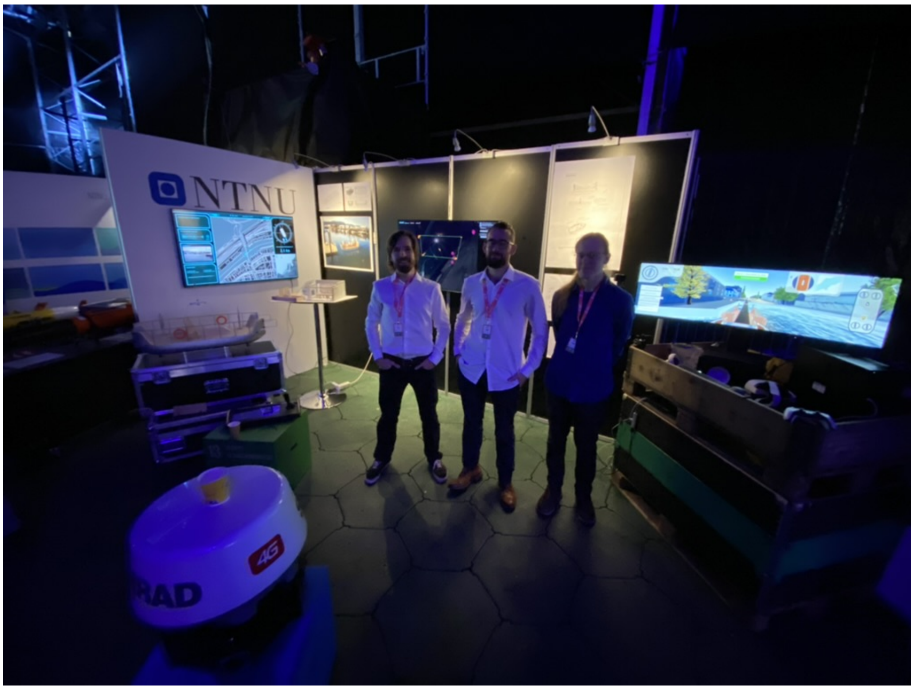


Figure 1: The simulator being showcased at UKATech 21

Figure 2: The simulator being showcased at NTNU Shore Control Lab

Contents

1 Introduction

In recent years, there has been a noticeable development within autonomy, AI, and autonomous transport. While it's most likely land-based vehicles that have gotten the most attention in the public eye, self-driving cars in particular, there is also considerable amount of ongoing research within the maritime domain.
A major part of this development is that of maritime autonomous surface ships (MASS). In 2018, the International maritime organization took the first steps to address the safety of autonomous ships, and defined MASS in the following way: "[...] a ship which, to a varying degree, can operate independently of human interaction"[8].
MASS is also mentioned in a 2021 UN report on sustainable transport: "The digital revolution is fundamentally changing shipping. [...] Most predictions are that autonomous or semi-autonomous operation would be limited to short voyages, for example, from one specific port to another, across a short distance."[5].

However, fully autonomous systems (without any supervision) is still far into the future. While emerging autonomous systems no longer require an on-board crew, there is still a need for a human in the loop. One solution that is seeing ongoing research, is having an on-shore control center (SCC), where a human operator monitors one or more autonomous vessels, ready to take over control when need be.

This thesis will focus on recent research on shore control centers in Norway, in the context of autonomous passenger ferries. This project is lead by NTNU, named the 'Shore Control Lab' (SCL). A central part of SCL's work, is the inclusion of a human operator. "The human remains "in the loop," and is always able to take preventive action if needed"[2].

However, testing of such an on-shore solution is a demanding task, especially for low-probability but high-risk situations that are either too costly, or too risky to test in real life. Therefore, the use of simulators is highly relevant for evaluating the safety of SCC operations.

Research topic The topic and goal of this master project is to explore the use of simulators and video games as a tool for research and prototyping of unmanned maritime vessels.

Research question Can video game technology be used to enhance early-stage research and prototyping of remote operation systems for unmanned (maritime) vessels?

1.1 Project structure

Initial development Development started as a part-time student assistant position, Spring 2021, where the author was tasked with developing a simulator that was to be used as part of PhD candidate Erik Veitch's experiments on SCC operators. During this time, a prototype was developed where two scenarios designed to test manual takeover could be played.
This project led to a collaboration with design student Erik Scott Hognestad, who at the time was working on designing a GUI for autonomous ferry remote control[4]. Part of Hognestad's prototype was implemented and added to the simulator by the author.
development is documented in sections 3.2 through 3.3.

Preparatory project The preparatory project focused on the use of video games and simulation technology for research and prototyping of unmanned maritime vessels. This was a continuation of the aforementioned simulator, where the goal was to develop a tool that would allow researchers to create their own scenarios within the simulator. This tool was named the 'scenario builder' internally.
The preparatory project was done in collaboration with Sondre Ek, who was working on his master's thesis at the time[6]. Ek created a visual prototype for the scenario builder, and participated in facilitating user tests and a case study regarding the scenario builder.
Development was done by the author, where Ek assisted with implementing the redesigned menu screen.

Development is documented in sections 3.4 through 3.5.

Master's project The master's project was a direct continuation of the preparatory project. Development was done by the author. Part of this development focused on monitoring multiple ferries at once, which led to a collaboration with master student Jooyoung Park[11]. At the time, Park was designing an interface for monitoring multiple autonomous ferries. Parts of her functional prototype was integrated in the simulator.

Development is documented in section 3.6.

2 Background

The simulator developed for this master's project is based on ongoing research on autonomous passenger ferries performed at Norwegian University of Science and Technology (NTNU). In this chapter, parts of this research will be presented.

Milliamperé 1 Milliamperé 1 is a half-scale prototype passenger ferry constructed in 2017 at NTNU, featuring motion planning and control, and collision avoidance. It uses radar and lidar (and other sensors) for obstacle detection and tracking, which were key for achieving automated situational awareness [2]. This ferry was not built for commercial use, and was instead designed to be used in research on autonomy (ibid.)

Milliamperé 2 Milliamperé2 is a full-scale autonomous passenger ferry, building upon the Milliamperé 1 project. While Milliamperé 1 was mostly aimed at research, Milliamperé 2 focuses more on the reliability that is required for continuous operation. It was launched June 2021, and as of May 2022, is an ongoing project. (ibid.)

Gemini simulator

> Gemini is a Unity-based visual simulator originally developed by graduate students at NTNU in Trondheim, Norway. The project began as a simulator for the Milliamperé Autonomous ferry. The purpose behind Gemini is to provide a foundation for EMR (Electromagnetic Radiation) based sensors, such as optical cameras, LiDAR and Radar for use in development and testing of autonomous applications. [1]

Shore Control Lab The NTNU Shore Control Lab (SCL) is a shore control center (SCC) owned and operated by NTNU. [12] It acts as a testing facility focusing on land-based operation of autonomous vessels, and is based on the premise that autonomy should not aim to completely remove the human, but rather have a human-in-the-loop approach. [2]
Research areas include situation awareness, user experience design, trust in autonomy, among others. (ibid.)

> Our vision is a future where humans and AI work together to enable more resilient autonomous systems. We work towards this vision by testing interaction solutions in our lab, which is connected to both a virtual simulator and real autonomous surface vessel. [10]

2.1 Malone's heuristics for designing instructional computer games

Part of the research topic focuses on the use of video games in particular. It is therefore relevant to discuss if/to what extent the developed simulator can be considered a video game, and the unique benefits it provides (if any). To this end, "Malone's heuristics for designing instructional computer games"[9] will be used.

Malone argue that "[...] the essential characteristics of good computer games and other intrinsically enjoyable situations can be organized into three categories: *challenge, fantasy, and curiosity*".

Challenge

Malone proposes that the idea of having a goal is central when defining games. "In a sense, the very notion of a 'game' implies that there is an 'object of the game.' " (ibid.) A game's goal is

closely related to skill, in that it provides opportunity to test a player's skill, but also gives context to how the skill should be used. It also helps to teach a new skill, as a skill can act as the means to achieving a goal (ibid.)
However, not all goals are inherently good or appropriate for all games. Malone mentions that simple games should make the game obvious, but should not be trivial. The player will likely lose interest if a goal or outcome is completely certain. "In order for a computer game to be challenging, it must provide a goal whose attainment is uncertain" (ibid.)

Uncertain outcome For a goal to be engaging, it should have an uncertain outcome. Malone presents four ways for achieving this:

1. *Variable difficulty level* By offering multiple levels of difficulty, a game can provide an appropriate amount of challenge to a larger range of players. This is achieved by (1) being determined by the program automatically, (2) having the player choose, or determined by an opposing player (ibid.)
2. *Multiple level goals* Even if the outcome of one goal is certain/trivial for some players, they can still be challenged by other, more complex goals. Malone calls this a 'metagoal', where the challenge is to reach the basic goal more efficiently. The two main types of metagoals are (1) score-keeping (using some meaningful performance metric), and (2) speeded responses, either performing something quickly, or before a timer runs out.
3. *Hidden information* A game's outcome can be made uncertain (and therefore more challenging), by selectively revealing information to the player.
4. *Randomness* Randomness naturally creates unpredictability.

Self-esteem Challenge and self-esteem are related aspects, in that success can increase self-esteem, and increase player engagement, thus wanting them to play more. However, if the challenge is too difficult, it may discourage the player to the point where they become disinterested and stop playing. It is therefore related with providing multiple levels of difficulty.

Fantasy

"In general, games that include fantasy show or evoke images of physical objects or social situations not actually present. "[9].
When playing a game, the player can enter a fantasy world where they assume the identity of a fictional character. Malone mention how fantasy can be used to increase the fun of learning, by having them solving a fantasy goal instead of presenting the learning goal directly.

Intrinsic and extrinsic fantasies A fantasy is extrinsic when the skills required to succeed are not a part of the game's fantasy. Malone uses Hangman as an example: The skill of word-guessing is in no way related to the man being hung.

A fantasy becomes intrinsic when the fantasy and skill depend on each other. This often means that the game has 'fantasy goals', meaning that the goal is presented as part of the fantasy world.

Emotional aspects of fantasy Emotions and fantasy is closely related, as it makes sense that a fantasy who immerses and resonates with the player is likely to create an emotional response. However, different people will react differently, and may prefer different fantasies than others.

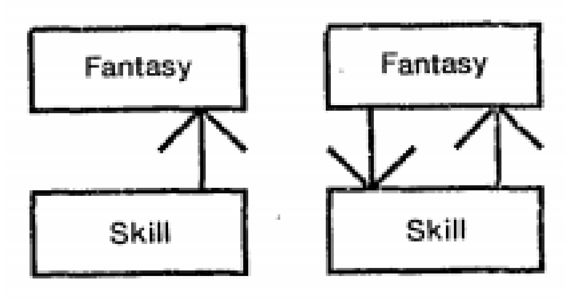


Figure 3: Logical dependencies in extrinsic and intrinsic fantasies[9]

Curiosity

Malone distinguishes between sensory curiosity and cognitive curiosity.

Sensory curiosity This involves using visual and auditory stimuli to attract attention, and to provoke curiosity. This can be done in four ways: (1) as decoration, sound and graphics that does not react to the player's action in any way, (2) To enhance fantasy, which is where the decoration becomes a part of the game's fantasy, (3) To reward good performance, which can accompany a challenge to increase player engagement, and (4) to represent information more efficiently than using plain text or numbers.

Cognitive curiosity This pertains to a player's desire to complete their knowledge about a game's underlying logic. "the way to engage learners' curiosity is to present just enough information to make their existing knowledge seem incomplete, inconsistent, or unparsimonious." (ibid.)
This can be done by providing surprising or informative feedback that is not seen as trivial to the player (based on their current knowledge of the game) (ibid.)

2.2 Simulation fidelity

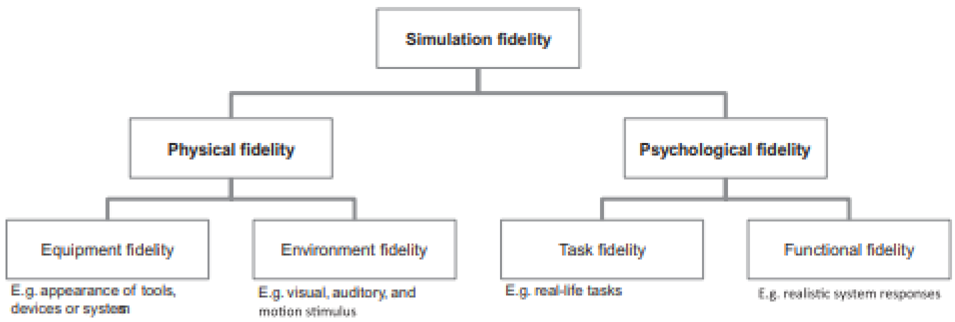


Figure 4: Simulation fidelity dimensions[3]

The simulation fidelity dimensions will be used alongside Malone's heuristics to discuss development and presented findings. In the scope of this project, only physical fidelity will be discussed.

Physical fidelity How closely a simulator replicate physical elements of its real-world counterpart[3], and is often divided into two subcomponents: Equipment fidelity and environment fidelity.

Equipment fidelity Refers to the equipment a participant uses to operate the simulator, and how closely it resembles the corresponding real tools, devices, or systems (ibid.)

Environment fidelity How well the simulator mimics the physical aspects of the real-world environment it aims to replicate.

3 Development

In this chapter, the simulator's development will be presented in its entirety. Each feature are presented roughly in the order they were implemented, where subsequent iterations/changes are presented later in the chapter. The order is not completely chronological, as related features have been grouped together to better convey the system as a whole. In reality, the development process was more iterative. In some cases, multiple features were implemented in parallel, and went through several iterations and revisions.

3.1 Development process

An iterative development and test approach was used, where all (planned) features were tracked using a kanban board. Features were moved in chunks from 'Todo' to 'In progress'. Then, once all features were marked as ready for testing, internal iterative tests were performed. The 'Todo' list was then updated accordingly, before beginning the cycle anew (cards were either approved or moved to either 'In progress' or 'Backlog'). Issues were tracked in a separate, but similar board.

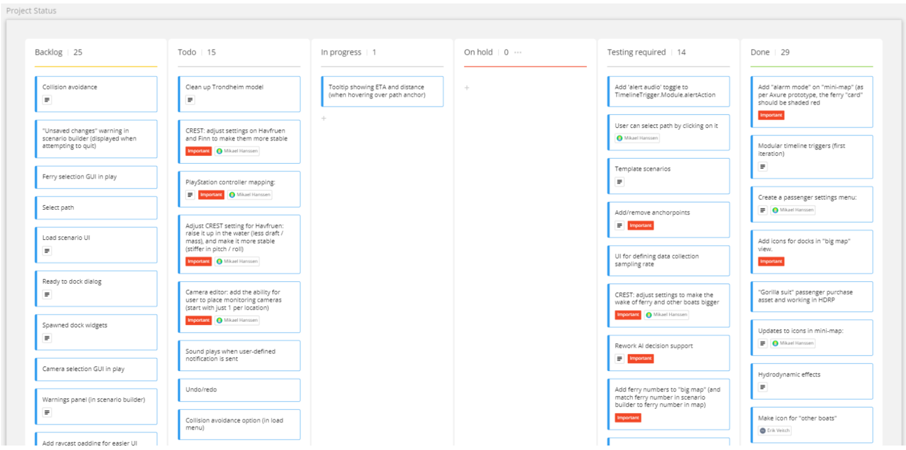


Figure 5: Kanban board for project status

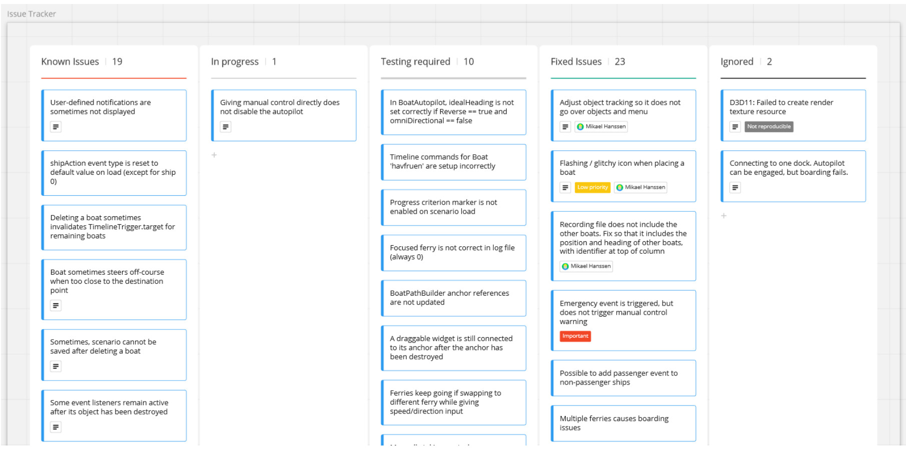


Figure 6: Kanban board for tracking known issues

3.2 Simulator V1

Development started as a scientific assistant project in collaboration with NTNU and Zeabuz, which at the time were researching human-machine interaction (HMI) for autonomous ferries.

The initial goal was to develop a simulator that was to be used in Zeabuz's research on HMI, more specifically focusing on unmanned maritime vessels. The existing Gemini simulator was used as a starting point.

It was decided that a minimum viable product (MVP) should be developed with two main parts: (1) autonomous ferry transporting passengers across a canal, (2) a scripted scenario where an operator overseeing the ferry must assume manual control following an onboard system failure.
As the implementation of the ferry's autonomous systems were not a main focus of this MVP, it was intentionally simplified. The goal of the MVP was to create a scripted scenario that was immersive enough to be suitable for user testing, not an accurate simulation of the ferry's sensors and autonomous systems. However, it remained of interest to integrate Gemini's detailed simulation of the ferry's sensors at a later date.

3.2.1 Autonomous ferry operation

The functional requirements for (1) were that the ferry should be able to...

1. mimic the movement of the real-life Milliamperé 2 ferry
2. approach and connect to its designated docks
3. handle passengers boarding/disembarking
4. cross back and forth between two designated docks
5. perform crossing fully autonomously
6. stop for crossing boats

Autonomous ferry and manual control The ferry's autopilot was simulated by having it follow a predetermined bézier curve path. Ferry movement was implemented using Unity's built-in physics engine (Nvidia PhysX engine)[1].

With the four-thruster setup of the real-life Milliamperé 2 ferry[2], it has a greater range of motion than a regular boat, allowing it to go sideways, rotate in-place, and drive in either direction. This was mimicked in the simulator by implementing three degrees of motion, two linear and one rotational. Figure 7 shows how this movement was mapped to the keyboard, along with other keybindings for controlling the ferry manually.

A simplified collision avoidance system was implemented, where the ferry would simply stop if a boat crossed in front of it (based on current heading).

[1] https://docs.unity3d.com/Manual/PhysicsOverview.html

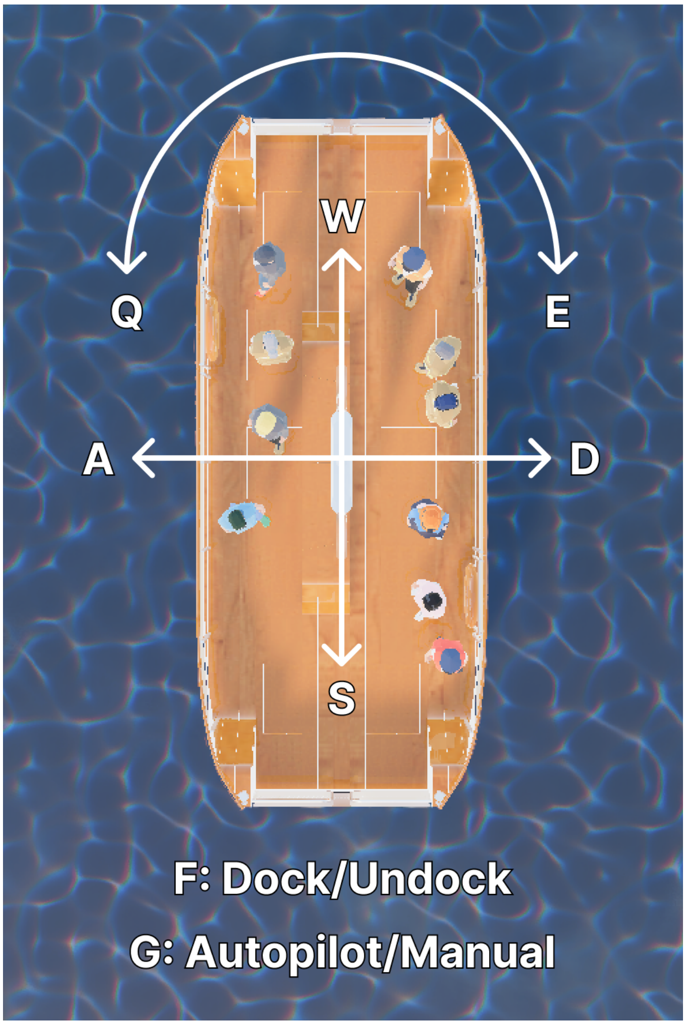


Figure 7: The ferry's three degrees of motion, with corresponding keybindings

Fully autonomous crossing By aligning each endpoint of the ferry's path to two docks placed in the scene, autonomous docking could also be simulated. The ferry's doors and latches were animated to open/close on docking/undocking.

Finally, passenger boarding/disembarking was implemented, using Unity's built-in pathfinding system[2]. The passengers were animated and given a randomized appearance using a 3rd party package from Unity's asset store, called "Advanced people pack"[3].

After making boarding start automatically when the ferry docked, and making the ferry undock automatically once boarding was completed, the entire ferry trip was now fully autonomous.

[2]https://docs.unity3d.com/Manual/Navigation.html

[3]https://assetstore.unity.com/packages/3d/characters/humanoids/humans/advanced-people-pack-2-170756

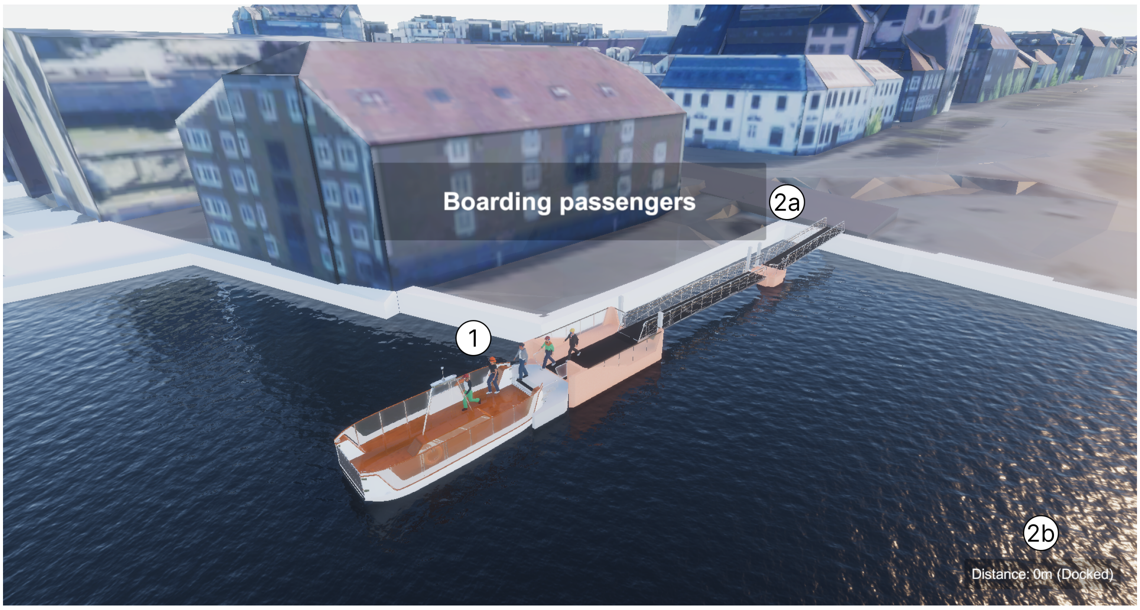


Figure 8: Boarding passengers

An explanation of figure 8 follows:

1. Ferry has docked autonomously and is now boarding passengers
2. (a) A message box describing what operation the ferry is currently performing. Possible messages are: “Boarding passengers”, “Boarding complete”, “Docking failed”, “Docking successful”, “Autopilot engaged”, and “Manual takeover required”.
   (b) A widget showing distance to the ferry’s destination dock, and whether it is docked or not.

3.2.2 Scripted takeover scenario

The functional requirements for (2) were:

1. A fully scripted scenario should be developed, which simulates a situation where manual takeover is required
2. Whenever manual takeover is required (as part of the scenario), the ferry should halt any ongoing actions, and give full manual control to the operator
3. The operator should be able to manually take over control at any time

As the simulator were to be used primarily for research, a key feature was having a set of predetermined scenarios designed to test an operator's ability to take manual control.

3.2.3 Exploring possible scenarios

Before implementing the first scripted scenario, the researchers at SCL were consulted regarding which scenarios would be of interest. From this, it was decided that the scenarios should simulate low-probability, but high risk events, where manual takeover would be required. A list of such events were created, which can be divided into three categories:

- System failures
  - Sensor malfunction (e.g. lidar malfunction)
  - Losing connection to/communication with the ferry (e.g. GPS signal lost)
  - Battery malfunction
- Passengers at risk
  - Passenger hurt during boarding/crossing
  - Passenger falling overboard
- Environmental factors
  - Extreme wind and/or waves
  - Thick fog

It was decided that the MVP would have a scenario involving a system failure onboard the ferry that would require manual takeover to resolve.

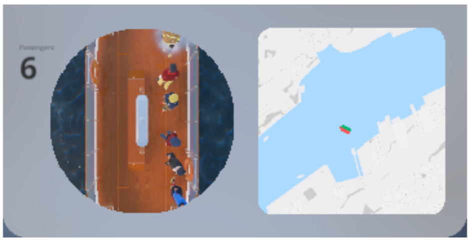
3.3 Simulator V2: Improved operator interface

3.3.1 Head-up display

After the manual takeover scenario was implemented, followed development of a head-up display (HUD), which is “any transparent display that presents data without requiring users to look away from their usual viewpoints”[4]. While a placeholder HUD was implemented in V1, it was severly limited (See figure 8).
This lead to a collaboration with Erik Scott Hognestad, who at the time was working on the design of an operator interface for monitoring multiple autonomous ferries, as part of a “Design 6” project[4]. See figure 9 for an explanation of each widget.

(a) Ferry dashboard

(b) Widget displaying the ferry’s current route, and current control mode



(c) Overview of the ferry’s passengers, and the ferry’s route

(d) Overview of the ferry’s thrusters and cameras

Figure 9: All HUD widgets

[4]https://en.wikipedia.org/wiki/Head-up_display

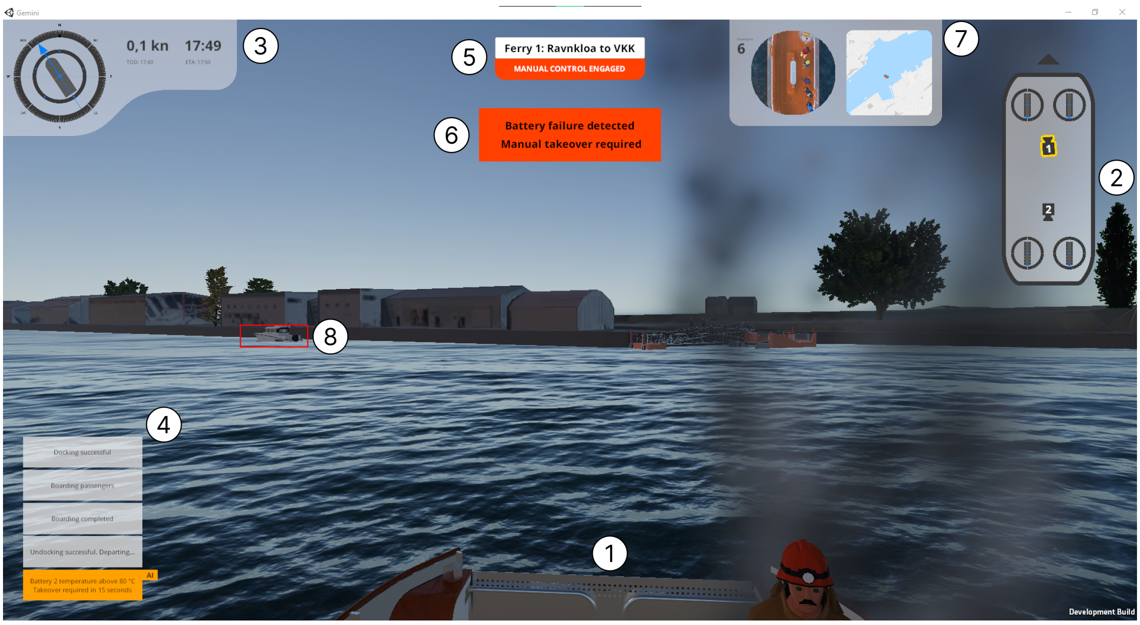


Figure 10: Playing a scripted scenario

An explanation of the user interface, as seen when playing a scenario in V1 (see figure 10):

1. Player views through camera onboard ferry (can be moved around)
2. A widget showing the current status of the ferry's thrusters and onboard camera
3. Dashboard showing the ferry's heading (compass), current time, speed in knots, TOD (time of departure) and ETA (estimated time of arrival).
4. The ferry sends messages to the player about the current state of the overall system, as well as what it is currently doing. Important messages are highlighted in yellow. Figure 10 shows a warning about a battery overheating. This feature was named 'AI decision support', and could be toggled on/off by the user.
5. Widget showing the ferry's planned route, as well an indicator for whether the ferry is in autonomous or manual mode.
6. A warning that is displayed after an emergency has occurred (with audio).
7. Dome camera showing onboard passengers (with passenger count), and a minimap of the ferry's surroundings.
8. Bounding boxes are displayed for all vessels not controlled by the player. Object detection can be toggled on/off by the user (See figure 23).

3.3.2 Data collection

As the simulator was to be used as a research tool, an essential feature was the ability to automatically collect data while playing a scenario. The following data was collected at a regular interval, and stored as a comma-separated file (csv):

- Remaining distance (distance from ferry to destination dock)
- The ferry's speed (in m/s)
- The ferry's world position (relative to Unity's world grid)
- The ferry's position relative to it's initial position (i.e. when the scenario began)
- The ferry's heading, as a normalized vector
- Linear input (as a 2D vector with range [-1:1])
- Angular input (decimal number with range [-1:1])
- Control mode (1 for manual, 0 for autonomous)

3.3.3 Alternate input devices

In the MVP, the simulator was controlled solely using a keyboard and mouse. While the keybindings shown in figure 7 may be familiar to operators who have previously played video games, due to the WASD layout used in many first-person video games[7], it is not directly intuitive. There is no direct connection between each key and how they operate the ferry, meaning that the operator is forced to train and memorize each keybinding before any scenario can be played.
This lead to the consideration of using other peripheral devices. Because of this, support for the Playstation 5 controller was implemented, so that the ferry could be controlled without the use of a keyboard. This was used in combination with a stream deck[5], which is a board of keys with LCD screens. Each button is customizable, and can be given descriptive icons to better communicate their function to the player. The stream deck was used for (1) switching between ferries, (2) which ferry camera should be viewed (either front or back), (3) toggle autopilot on/off, and (4) docking/undocking

Figure 11: Button mappings for playstation controller

Source: http://playstation.com (background image)

3.3.4 Use of ambient and functional audio

Ambient audio of waves and seagulls was added, with the aim of increasing immersion while playing. Some functional audio was also added:

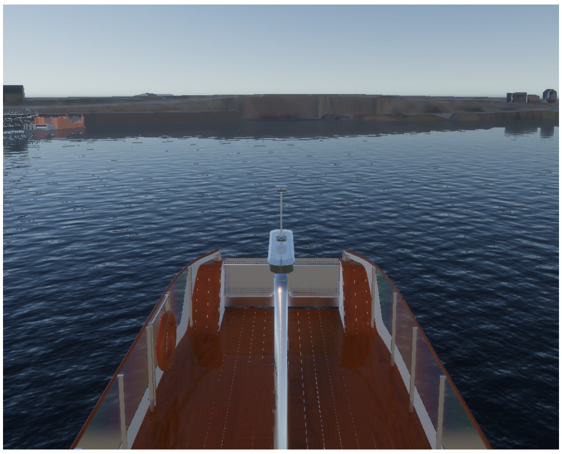
[5]https://www.elgato.com/en/stream-deck

- Motor sound where the pitch was changed based on the ferry's velocity
- A bell ringing twice when ferry starts to cross (i.e. undocks)
- Collision sound, where the volume was scaled based on the impact velocity

3.3.5 Hydrodynamics

While the implementation of the ferry's motion gave the player direct control of the ferry (when required), it was lacking in environment fidelity. The movement was essentially locked to a two-dimensional plane, without any of the physical properties one would expect from moving through water. It was therefore decided to make use of the Crest ocean system[6], an external package available in Unity's asset store. This package in particular was chosen, as it was previously used in the Gemini simulator, and has the following benefits:

- Detailed ocean visuals, with three-dimensional, dynamic waves
- Designed to work alongside Unity's built-in physics engine
- Simulates buoyancy and drag through water



(a) Old visuals; an ocean shader applied to a plane

(b) Crest ocean system

Figure 12: Ocean visuals comparison

3.4 Scenario builder V1: MVP

The initial development of the simulator sparked an interest in exploring a more general usage of simulators and video games in research and prototyping, and that the simulator should be expanded upon to be used in a case study towards this area of interest. Following conversations with the research team at SCL, it was decided that early development should be aimed at researchers that have an interest in using the software described.
During development, having only scripted scenarios was found to be too restrictive for the iterative process of development that follows from early-stage research and prototyping. From this emerged the need for a tool where researchers could define and alter scenarios as needed. Due to its relevance, it was decided that the development of a scenario builder tool should be the main focus of the preparatory project, leading up to the master project.

[6]https://assetstore.unity.com/packages/tools/particles-effects/crest-ocean-system-hdrp-164158

3.4.1 Functional requirements

Before development began, a set of functional requirements were made for a scenario builder MVP:

1. Same as the simulator, the scenario builder should be designed to be used primarily as a research tool.
2. The simulator should provide a visual, low-code interface, so that it can be used by researchers with little/no programming experience.
3. The MVP should provide the tools necessary to fully recreate the scripted scenarios implemented in the previous version of the simulator.

Additional functional requirements were also given for the scenario that was to be used in Case 1 (section 4.2.2):

1. The scenario should have the following takeover event: There should start a fire onboard the ferry, forcing the test participants to take control and manually dock.
2. Some boat traffic around the ferry, so that they have to avoid collision while driving manually.
3. The scenario has to be identical for all participants, with the exception of when the takeover event occurs (as described in the next point).
4. The ability to specify how long the participant have to wait before the takeover event occurs.
5. When the takeover event occurs, there should appear a centered alarm message. It should blink and have a relatively loud alarm sound, with the ability to toggle the sound on/off.

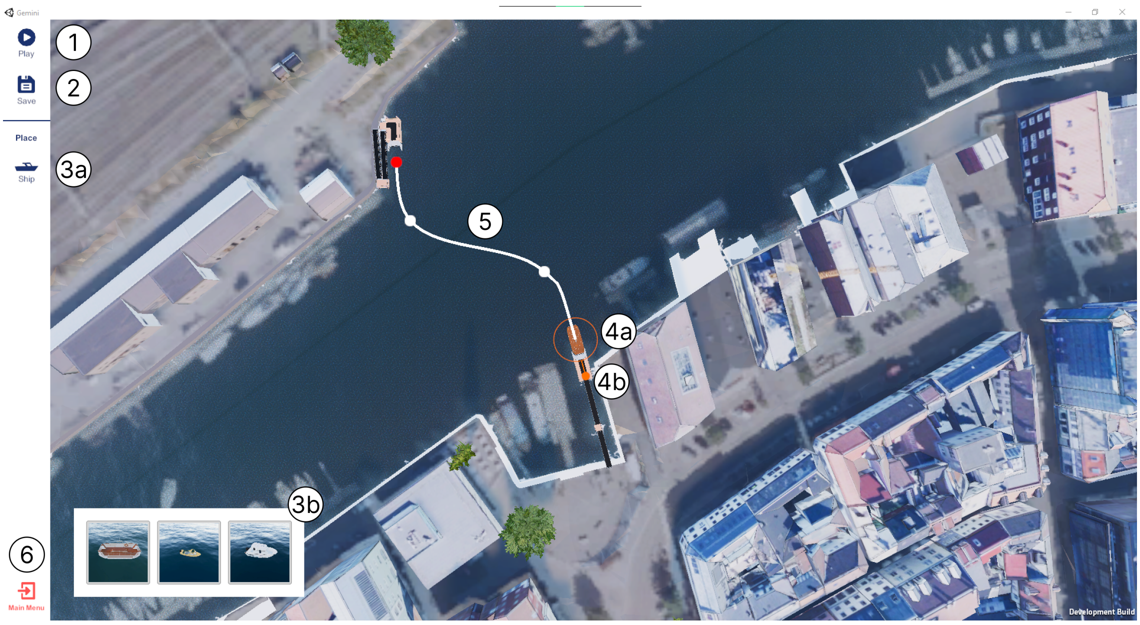


Figure 13: First version of the scenario builder interface

1. Preview scenario by playing it (Changes are saved before playing)
2. Save scenario manually
3. (a) Shows/hides menu for spawning boats
   (b) List of available boats. When clicked, the respective boat is spawned in the screen's center
4. (a) Move boat with drag and drop
   (b) Rotate boat with rotation anchor
5. Alter the boat's path by dragging the anchor points (destination is marked red). Number of anchor points are fixed.
6. Return to main menu

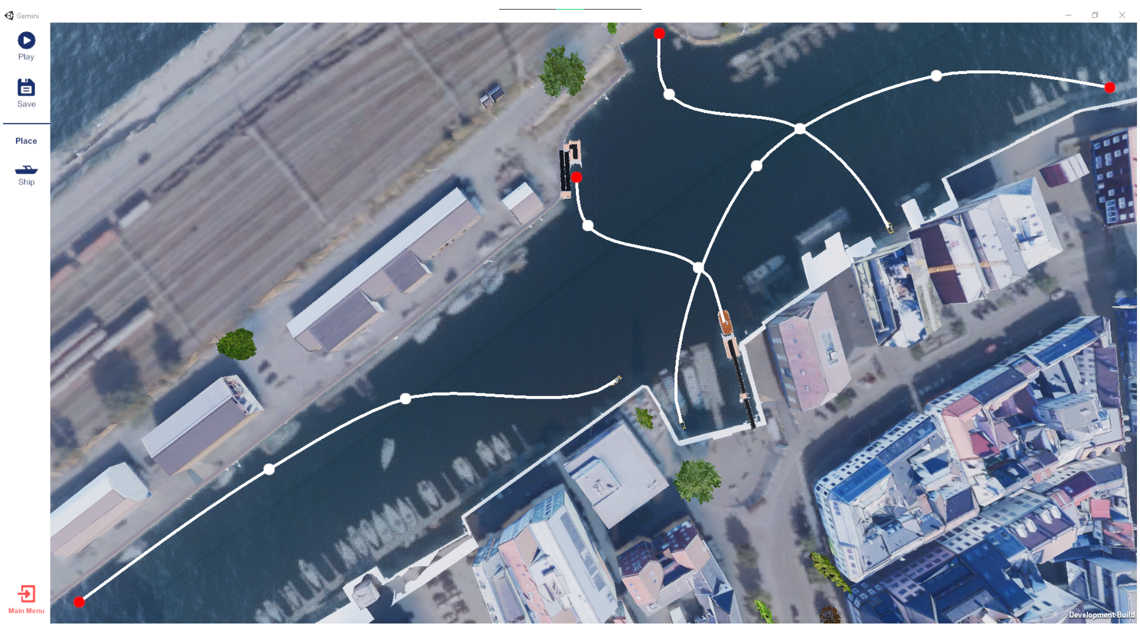
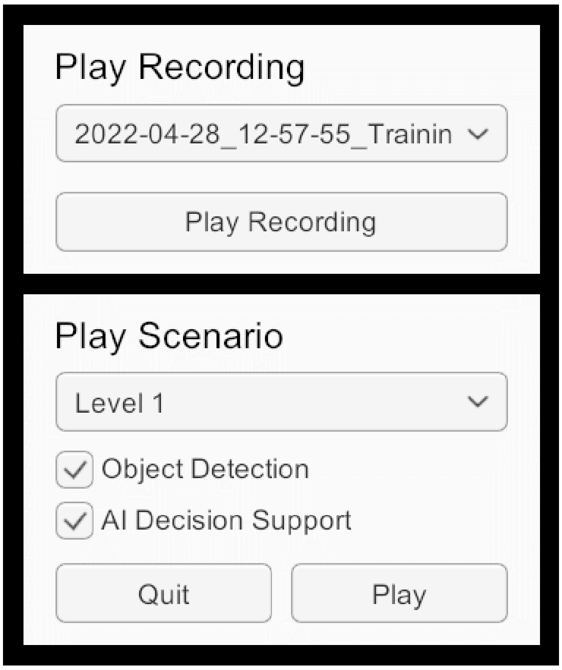
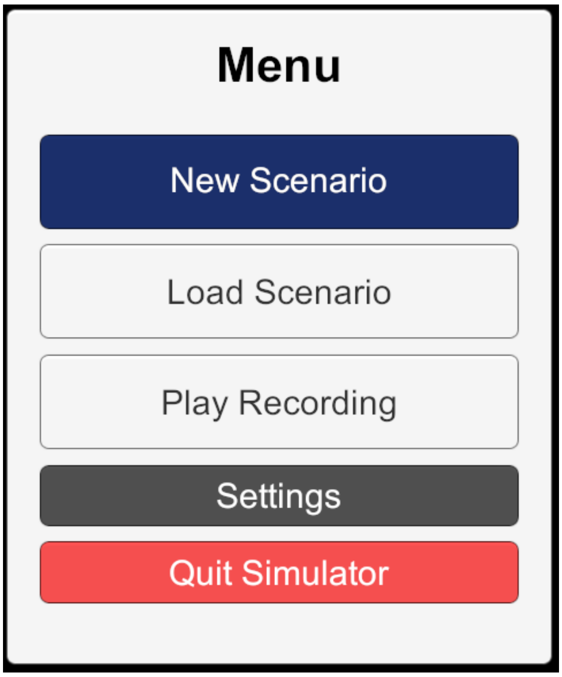


Figure 14: Scenario with multiple boats

3.4.2 Redesigned main menu



(a) Old main menu



(b) Redesigned main menu

TODO: Refer to Ek's work, and present a summary for design choices.

3.5 Scenario builder V2: Custom scenario logic

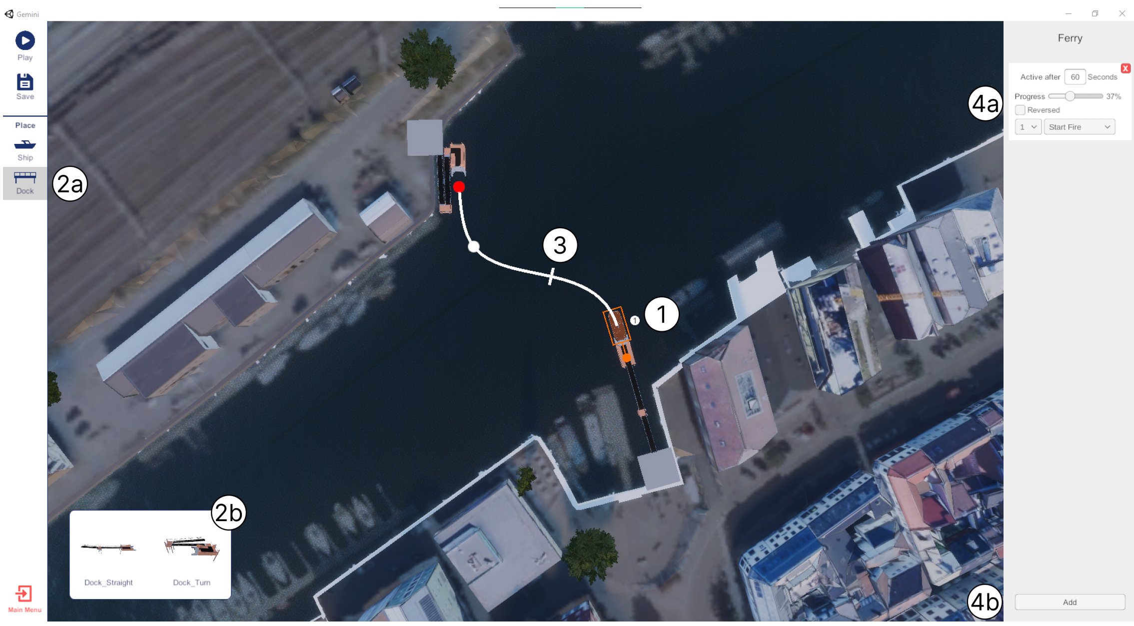


Figure 16: First version of event triggers, used to define scenario logic

1. Each boat in the scene is given an unique ID number (here the ferry is 1).
2. Menu for spawning docks. Same as with boats, docks can be moved and rotated (when selected)
3. A marker corresponding to the trigger's 'progress' slider (See 4a)
4. (a) Action trigger (Also see figure 17)
   (b) Button for adding a new event trigger

Action triggers Action triggers allow the user to activate a specific action once a set of criteria are met.

Figure 17: More detailed look at an event trigger

1. Each action trigger has three criteria. If all are met, the specified action will be performed once.
   (a) The trigger will remain inactive until the given time has passed.
   (b) Boat must complete the given percentage amount of its journey.

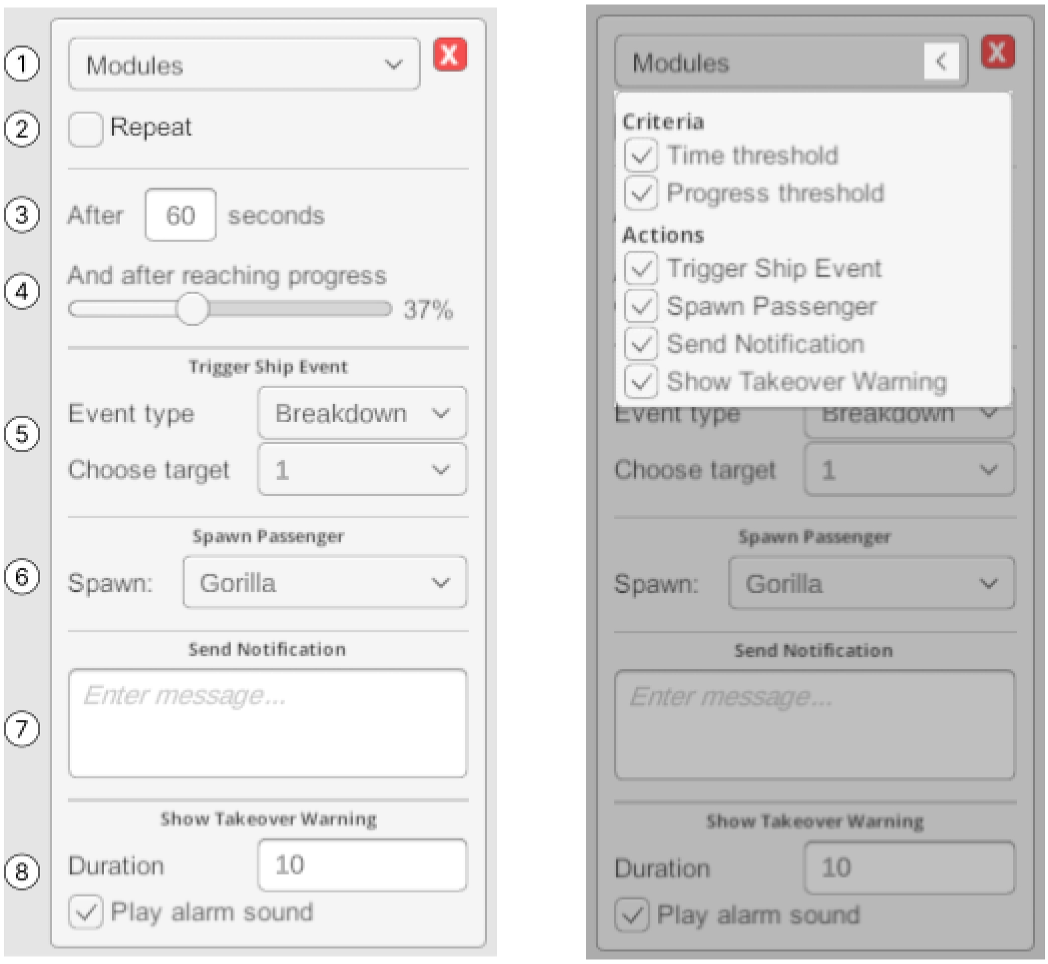
(c) If enabled, the event will only be triggered when the ferry is going in reverse (i.e. during the return trip).

2. Specifies which action should be performed when triggered, and which boat should be targeted (using its ID number).

3.6 Simulator V3: Current version

This section describes the development done during the master's project, and is a continuation of the preparatory project.

3.6.1 Increased modular logic



(a) All available modules

(b) Menu for selecting modules (Highlighted)

Figure 18: A more modular event trigger

Event modules Similar to the first action trigger implementation, all specified actions are performed once all active criteria are met.

1. Shows module list when clicked (See figure 18b)
2. If enabled, the trigger will be repeated after activation (all criteria are reset).
3. Trigger waits for (at least) the specified amount of time

4. If active, the event will be triggered when the boat crosses this point in its path (if all other criteria are met).
5. Specified action is triggered on the specified target (targets the selected boat by default).
6. Spawns the specified passenger on the current destination dock (this is done to ensure the player doesn't see the passenger spawning)
7. Displays the message as a notification to the player (with a highlighted color).
8. Displays a red blinking warning, telling the player to take control. It will be visible for the specified duration, with a countdown and alarm sound (if enabled)

Design changes An user test performed for the scenario builder (findings presented in section 5.1) revealed that each element of the action trigger did not clearly communicate its function. Each element were grouped together as a single block, with no clear separation between the event's activation criteria and actions. To address this, the event trigger was turned into a collection of physically separated modules. A separate menu was added (see figure 18b), where modules where clearly separated into 'criteria' and 'actions'. Modules not selected by the users are hidden. Selected action modules are separated with a horizontal line and its name. Criteria are still grouped as a single block, to convey the fact that all criteria must be satisfied at the same time, before any action are triggered. When they are all met, all selected actions are performed once.
An option to repeat the trigger was added. This causes all criteria to reset when the event is triggered.

3.6.2 Increased available information for the operator

Once support for controlling multiple ferries at once was implemented, a higher amount of information had to become available to the user, where the goal was to give a quick and comprehensive overview of the ferries and surrounding sea traffic at all times. This lead to a collaboration with master student Jooyoung Park, who at the time was working on interface design for remote operation of multiple ferries[11], with SCL in mind.

- Support for multiple displays
- Terrain/sea map showing all vessels in a given area (See figure 22)
- More detailed information for each active ferry (See figure 21)

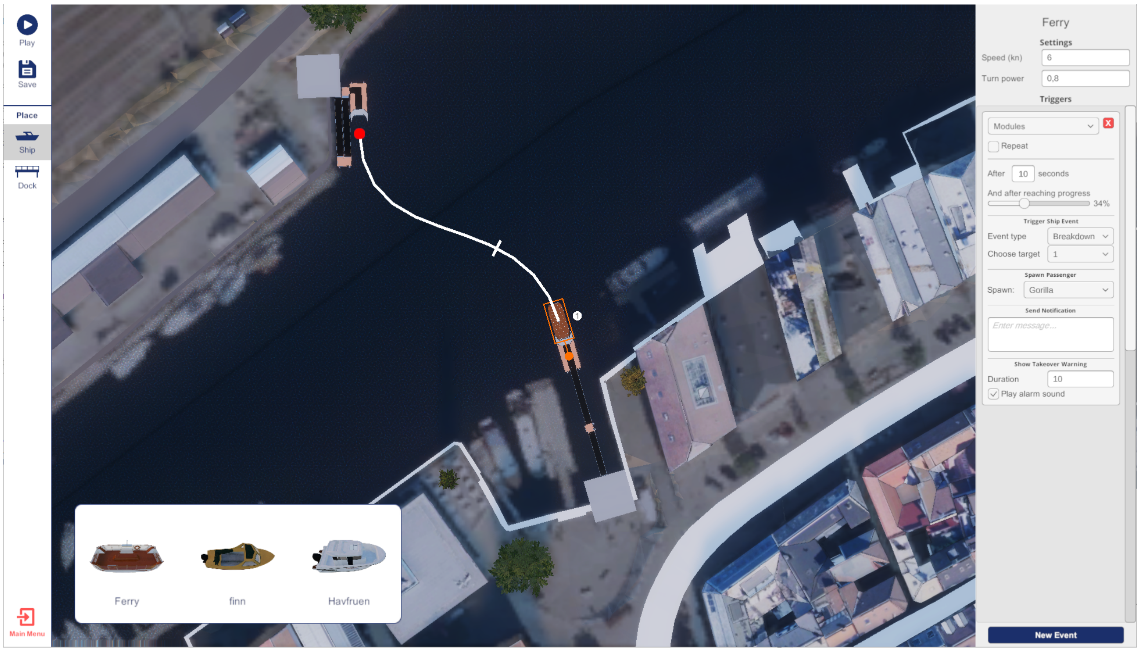


Figure 19: The scenario builder

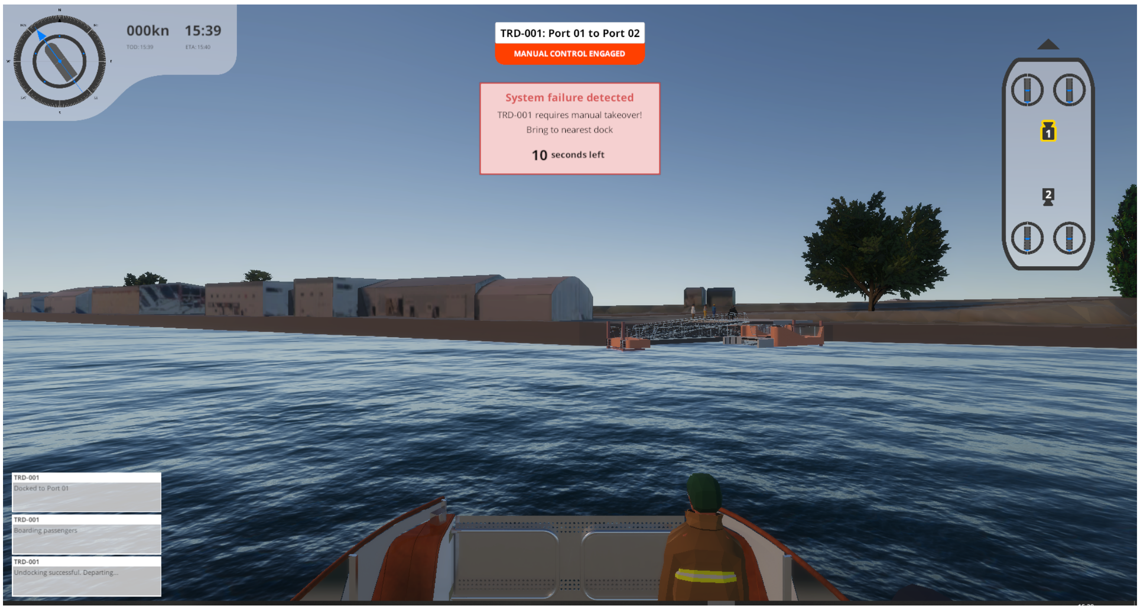


Figure 20: In-game takeover warning

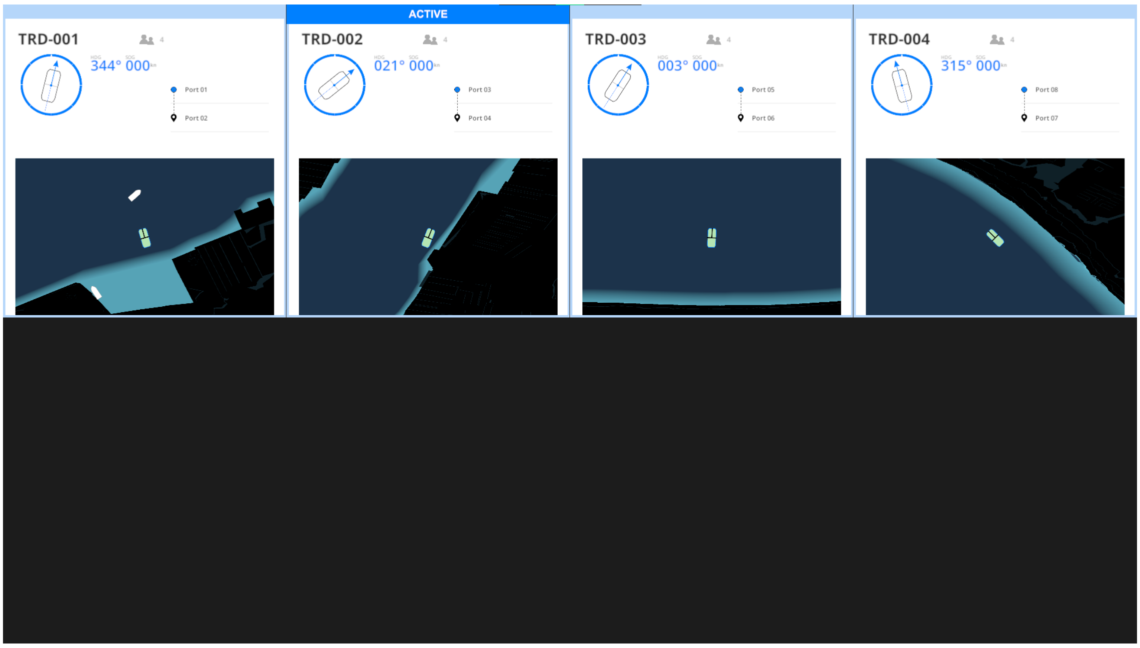


Figure 21: Detailed info for each ferry, including mini-map
(can display up to 8 ferries at once)

Figure 22: Sea map giving an overview of all traffic in the area

3.6.3 Automatic data collection

While playing a scenario, data is collected at a regular interval (which can be configured by the user, see figure 23). This includes:

- General data for all boats (position and rotation)
  - Stored as a recording that can be replayed in-simulator
- Detailed info for ferries, saved as a single csv file

- Speed (m/s)
- Heading (degrees)
- User input (two linear axes, one rotational)
- Autopilot state (on/off)
- Distance to destination
- Camera currently in use (onboard ferry)
- Which ferry the player is looking at
- Docked state
- Collision log (separate file)

- Unix timestamps are included with all measurements, so that it can be synchronized with additional (external) data, if need be.

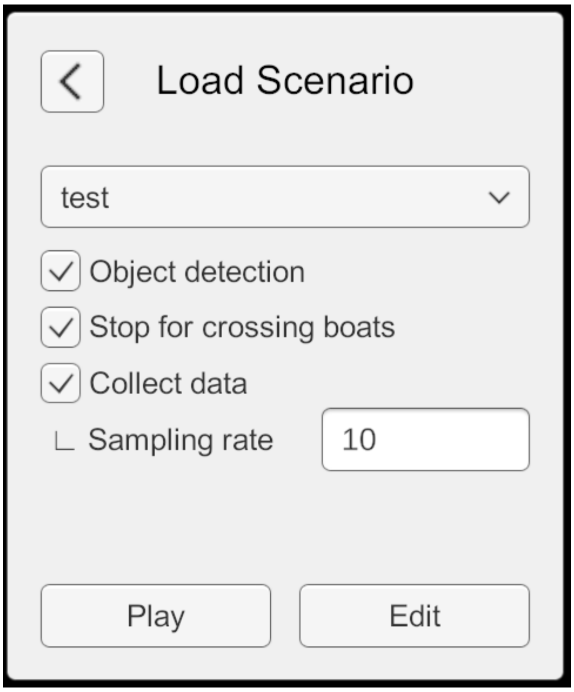


Figure 23: Data collection, object detection, and obstacle avoidance can be toggled from the load menu

4 Method

In this chapter, the methods for testing the software, and how researchers used the resulting tools for their own experiments, will be presented. Internal tests were performed during development as part of an iterative development cycle, described in section .

4.1 Exploratory testing

During the early stages of development, tests were open-ended and exploratory in nature. During the development of the first MVP, tests were mostly performed to evaluate the software, to uncover issues/bugs, and to get feedback from the researchers at SCL (based on the functional requirements they gave earlier). However, these tests also became an opportunity to explore new ways the simulator could be used, and to discuss possible new features that could be implemented in the future.

4.2 Case study

It was decided that a case study should be performed to further explore the relevance of the simulator, and to explore which features actual researchers require when using such a tool.

The purpose of the case study is to explore specific research areas where the simulator is applicable, and in that way implement more and more relevant features. Both cases should be presented with this goal in mind.

4.2.1 Selecting a case

When selecting a case, the following requirements were outlined:

1. The case should target a researcher/student who has expressed an interest in using a simulator as part of their research/study.
2. The case participant's area of research should be within HCI
3. A tool should be developed that will allow the case participant to create scenarios of their own design.
4. The simulator (and the scenarios created) should be used in an independent experiment/study, led by the case participant.

4.2.2 Case 1: Pilot study

Why this case was chosen Case 1 took place during the preparatory project leading up to the master project. The focus of this case was chosen to be on Kristin Grønhaug Senderud's studies towards attention span. This study was chosen, as (1) Senderud expressed an interest in using the simulator to test SCC operators (2) The research topic was on the attention span of SCC operators (3) Senderud gave functional requirements for a specific set of scenarios, which were used to guide the development of the scenario builder MVP, and (4) the pilot study was an independent experiment led by Senderud.

Preliminary test Before the pilot study, a preliminary user test on the scenario builder was performed, with Senderud as test subject. Sondre Ek was main facilitator during the test, with the author providing technical guidance where necessary.
Before the test began, the test subject was briefed by the main facilitator.
The following brief was used (translated from Norwegian):

The software we'll be using today is still in early development, so many bugs and issues are to be expected. Therefore, we may have to intervene during the test.

Given that the simulator is somewhat limited, I will be asking you to build the scenario in a specific order. We therefore have a set of predetermined task we would like you to perform. Together, these tasks should result in a scenario that closely match the original scenario you specified. After tasks these have been completed, you can explore freely.

Then, the following tasks were given:

Task 1: Create a new scenario with a fitting name

Task 2: Create and move out two docks, one of each type

Task 3: Create and move a ferry

Task 4: Configure so that a fire will appear on-board the ferry midway between dock 1 and dock 2

Task 5: Play the scenario

Task 6: Return to the editor from play mode

Task 7: Go back to play mode and try to manually control the ferry

Task 8: Place a new boat, and make it drive on a collision course (with the ferry)

Note that the initial plan was to perform this user test on-site at SCL. However, it was performed remotely on another computer, due to scheduling constraints.

4.2.3 Case 2: SCL Experiment

This experiment was a larger-scale round of experiments focusing on the role as a SCC operator. This experiment was planned and performed by researchers at SCL, led by PhD candidate Erik Veitch.
As including the experiment in its entirety as a single case would be outside the scope of this thesis, it will focus on the first round of experiments, and later use the interview answers as a basis for findings.

Recruitment For the first round of experiments, 16 gamers were recruited.

Tasks given Two scenarios was used for testing. In both scenarios, the operator was tasked to monitor a ferry autonomously crossing between two docks, waiting for a possible situation where they are required to take manual control. The scenarios had the following scripted takeover events:

Scenario 1: A system failure on-board causes the autopilot to shutdown automatically, requiring the operator to drive to the nearest dock before a timer runs out.

Scenario 2: Two boats begin to drive on a collision course towards the ferry. While the ferry stops for the first boat, the second boat will collide if the operator does not manually take over control.

The experiment further divided these scenarios into four factors. While these factors will not be explored in detail in this thesis, it is worth mentioning, as it meant that each test participant got a slightly different permutation of the two scenarios, based on these factors. However, each respective takeover event was identical for all participants.
These factors were (1) time until takeover event, (2) amount of active ferries (3) time available for taking control, and (4) whether the participants was warning about possible takeover while playing

4.2.4 Test location

As the simulator was primarily developed to be used as a research tool at SCL, all tests were performed at SCL, so as to keep the test environment as close to the actual research environment as possible. The only exception was the preliminary user test performed during Case 1, which was performed remotely.

Equipment used A widescreen was used for viewing the ferry's onboard camera, and to control the ferry. Additional information was displayed on the wall-mounted displays.
Different peripheral devices were also utilized during testing, namely a ps5 controller for operating the ferry, and a streamdeck for selecting camera, which ferry should be visible (on the widescreen), toggling autopilot on/off, and docking/undocking.

SCL's room layout The test room features an operator station with powerful hardware and a multi-display setup where the simulator was played (see figure 24). The lab also has a separate instructor station (not shown), where researchers can control the computer, monitor the test room via CCTV, and use two-way audio communication.[12]

Figure 24: The simulator in use at SCL's operator station

5 Findings

In this section, the findings from each case will be presented.

5.1 Case 1: Pilot study

Spawning, moving, and rotating dock seemed to be intuitive, but orienting it was not, as it was not obvious how the ferry would connect to the dock.

- **Create and place two docks**: Spawning, moving, and rotating dock seemed to be intuitive, but orienting it was not, as it was not obvious how the ferry would connect to the dock. It was not immediately obvious that anchor points along the ferry's path was movable, and that they should be used to alter the ferry's path.
- **Create and place a ferry**: Placement was obvious. Once interaction with anchor points was understood, connecting the ferry to a dock was also understood (i.e. dragging either start or end anchor points to dock).
- **Set a fire to occur on the ferry**: This task showed that the current way of defining triggers was not obvious.
- **Play scenario**: Entered play without issues. However, while playing, controls was not obvious, as it was primarily done via keyboard, but keybindings was not intuitive.
- **Setup another boat, and make it drive**: While setting up the boat's path was intuitive, configuring the logic for making it drive was not (for the same reasons as mentioned in the previous point). This was likely due to a lack of behaviour controls for the ships.

Note that these findings are from the preliminary test performed prior to the pilot study (during the preparatory project).

5.2 Case 2: SCC operator experiment

The simulator was used by a group of researchers at SCL to create a set of scenarios, which was then used in an independent experiment, led by said researchers. They also conducted a post-experiment interview, where part of the questions are directly relevant to further development of the simulator, and is therefore presented here.

5.2.1 Demographic data

All data presented in from the interviews during the first round of experiments at SCL, which recruited 16 test participants that had some form of "gamer" background.

Gender	3 female (18.8%), 13 male (81.2%)
Age	Between 22 and 41. Average: 24.94
Played ship simulators?	4 out of 16 (25%) had played some kind of ship simulator before.
Has boat license?	2 out of 16 (12.5%) has a boat license.
Maritime experience	10 out of 16 (62.5%) has experience with maritime activities.

5.2.2 Interview answers

Q3 *What information was useful when you were monitoring?*

In the answers for this question, there was a tendency for talking about the usefulness of each screen. Therefore, answers have been grouped into three binary factors, and one categorical (presented below).
Note that if an answer for a specific question did not explicitly answer one of the following factors, they were instead counted as "no answer". *Data is presented in figure 25.*
Selected factors:

(a) Whether they said the camera view was useful.

(b) Whether they said the ferry overview screen was useful.

(c) Whether they said the map screen was useful.

(d) Which screen they mentioned was most useful.

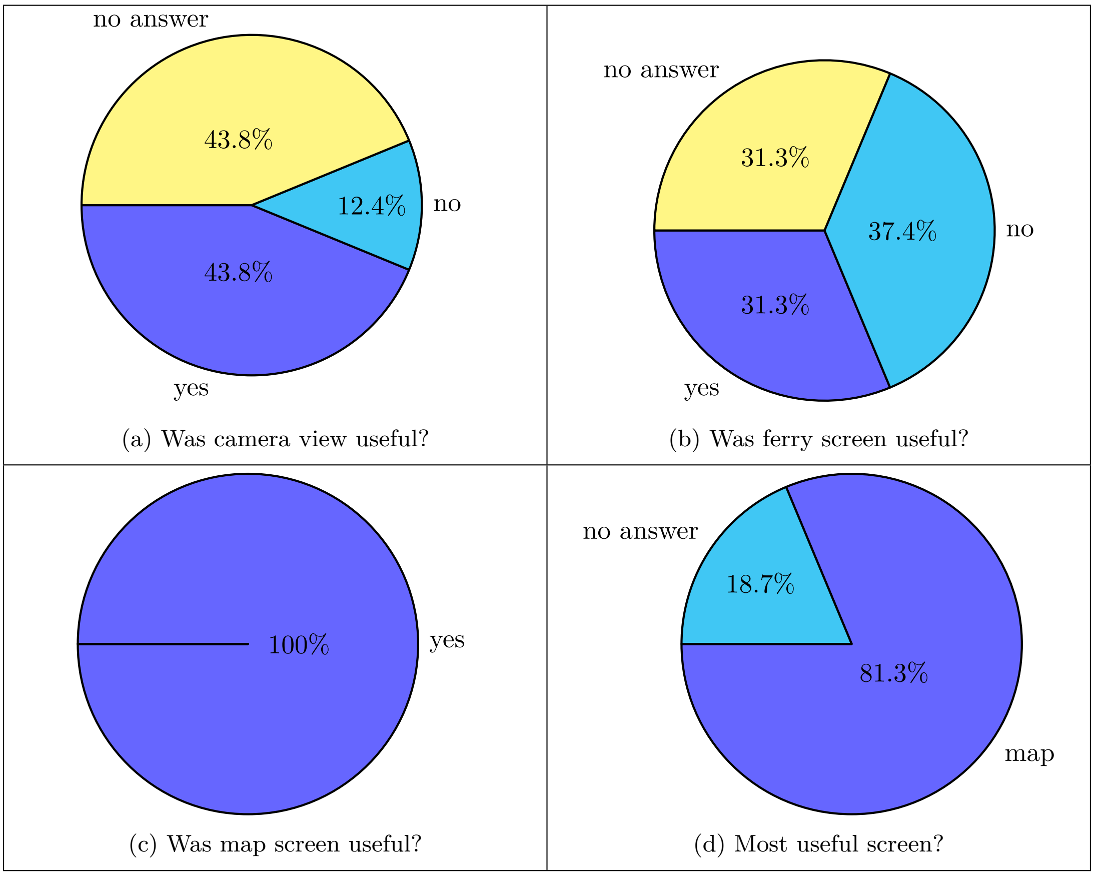
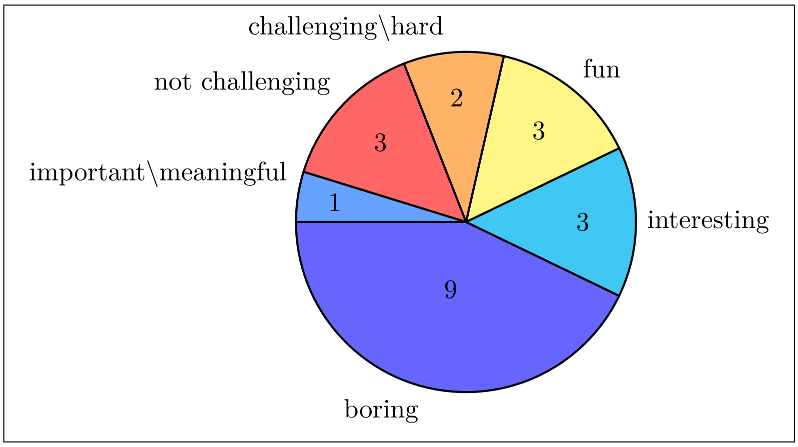


Figure 25: Which screens participants found useful

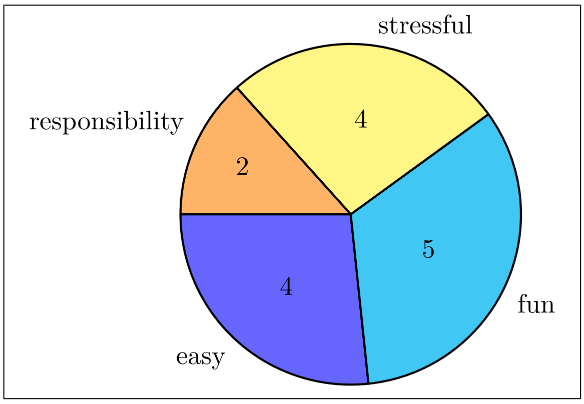
Q9 *How would you describe your experience as a Shore Control Center Operator?*

For this question, there was a split between participants finding the experience boring, and finding it interesting. The main reason that kept some participants interested throughout, seemed to be the novelty of the simulator and the role as a SCC operator.

Data coding The answers were coded into a set of descriptive phrases. This was done by making a list of all descriptive words in each answer.
Similar phrases were grouped together, such as synonyms (e.g. challenging/hard), and words with the same root (e.g. boring/bored).
The context was also taken into account. For example ‘not challenging’ should NOT be counted as ‘challenging’.



Q10 *How did it feel to take over control?*
The answers here have been coded into a set of descriptive phrases, using the same approach as in Q9.



Q14 and Q15

Q14: *Think back to the display. What could be improved? Was there something you felt could help you to have even better situation awareness?*

Q15: *What indicators might let you know that something's amiss with your vessel?*

In Q14 and Q15, the participants proposed specific changes and features they would like to see in the simulator. As these answers are related, Q14 and Q15 have been combined into a single list of features. Similar features have been grouped into relevant categories:

Map

- Grey icons for boats standing still
- Highlight currently selected ferry
- Highlight currently selected
- Field of view cones for ferries
- Line extending from ferry icon to better show heading and movement
- Proximity warning
- Predicted path/position for the ferry and other boats, based on their current heading and speed.

Minimap

- Have minimaps more zoomed out/change based on speed of incoming boats
- Show arrow on minimap for indicating boats approaching off-screen

Camera

- Single camera, with button to look in driving direction
- Remove the camera's rotational constraints
- Make camera face river at scenario start

- Increase field of view
- Button for aligning camera with driving direction
- Button for switching camera mode

Operator HUD

- Notifications
  - Issue: Grey text on grey background
  - Symbols for specific actions
  - Color coding (for repetitive messages)
- Have the compass on the top center of the screen
- A control panel with engine temperature, oil level, etc.

Situation Awareness

- Pop-up notification with button when a situation occurs, on every screen
- Proximity alert
- Audio
  - Microphone on-board ferry (under deck), so that the operator can hear potential issues with the engines.
  - When takeover occurs, have an auditory message on-board explaining to the passengers that a takeover is occurring, and why.
- Wind and current indicators

Manual Control

- Automatically remove autopilot when trying to move the boat
- Have to hold controller with both hands, more useful with keyboard
- Maybe joystick (for movement), keyboard for the rest
- Giving the ferry a "boost" when an evasive maneuver is required (within the speed limit)

Screen Layout

- Move screen with ferry overview physically closer, and/or make the content bigger
- Integrate map on large screen, instead of splitting the main screen

Simulated Environment

- Simulate wind and current to increase difficulty when controlling manually

6 Discussion

6.1 The simulator as a video game

Following the collaboration with Sondre Ek during the preparatory project leading up to this master project, he presented how the simulator should not be considered a game[6]. He argued that while a clear win state is a defining quality of games, it is detrimental if a game designer pre-defines a win state, as this is a decision that should be left entirely to the researcher using the tool. (ibid.) This is a fair argument when applied to the scenario builder, with the context of research in mind. It is true that such a tool should not be considered a game for researchers.

However, it is still relevant to discuss whether a test participant playing a specific scenario or set of scenarios will experience the simulator as a game, and whether this comes with any unique benefits.

6.2 Discussion structure

Malone's heuristics[9] for designing instructional games will be used as a framework to evaluate the simulator, using the findings to discuss which features of the heuristics are included.
This will then be used in combination with relevant parts of the interview answers (from section 5.2.2), to propose a set of functional requirements for the next iteration of the simulator.

For brevity, the interview answers presented in section 5.2.2 will be referred to as 'Qi', meaning the i-th interview question. The SCL experiments will be referred to as 'Case 2'.

When a researcher uses the scenario builder to create a specific scenario, they may define one or more player goals. Because these goals are defined entirely by the researcher, it becomes difficult to discuss the quality of all possible goals in general. Therefore, for the sake of discussion, the two scenarios created for Case 2 will be used as examples, and will be referred to as 'scenario 1' and 'scenario 2' respectively. These scenarios are described in section 4.2.3.

6.3 Simulation fidelity

Before applying Malone's heuristics, the physical fidelity part of simulation fidelity will be discussed here, which will then be used later in the discussion.

6.3.1 Equipment fidelity

The test location at SCL greatly increases equipment fidelity, as it is built to resemble how an actual shore control center might look like (as shown in section 4.2.4). The use of custom peripheral devices also increases fidelity somewhat. However, the use of a PS5 controller likely does not resemble the equipment used for actual remote control of autonomous vessels. Finally, the two-way audio communication system between test participant and researcher also resembles how an actual supervisor and SCC operator might communicate.

6.3.2 Environment fidelity

The simulated environment is a detailed recreation of Trondheim canal, a real-life location that has been used to test the Milliamperé prototype ferry in the past. Additionally, the Milliamperé project began with the ambition to replace municipal plans for an additional bridge between 'Brattøra' and 'Ravnkloa', two locations near the canal.[2] This fact, combined with the detailed 3D

city model, creates high environment fidelity, which more closely simulates the experience of an SCC operator.
The use of the Crest system for ocean visuals and hydrodynamics, as mentioned in section 3.3.5, is relevant here as well.

According to Malone “[...] the essential characteristics of good computer games and other intrinsically enjoyable situations can be organized into three categories: challenge, fantasy, and curiosity.”[9]

6.4 Challenge

Does the simulator provide a specific goal?
For both scenario 1 and scenario 2, the player is given an overarching goal: ‘Monitor the ferry/ferries crossing the canal. Look out and be prepared for possible situations where manual takeover is required’.
However, the exact failure states and win states of each scenario is not revealed.

For scenario 1, the win state is to reach the shore and successfully dock before the time runs out (i.e. the ferry becomes inoperable due to system failure). The failure state is not reaching a dock in time. This is a ‘hard’ failure state, in that the scenario ends immediately when the time runs out.
For scenario 2, the win state is to avoid the incoming collision. The failure states are to either (1) fail to spot the boat approaching on a collision course, or (2) notice the boat, but fail to move out of its way in time. This is a ‘soft’ failure state, as the player can still dock after a collision occurs. In this way, a partial win can be achieved.

How is the goal presented?
As mentioned, the player is given an overarching goal. This goal, along with win states and failure states, are not presented as abstract goals, but rather using the context of the in-game fantasy. In other words, the player wins by fulfilling their duty as a SCC operator, and by ensuring all passengers remain safe at all times. This is what Malone calls a ‘fantasy goal’: “The best goals are often practical or fantasy goals [...] rather than simply goals of using a skill”[9]. For scenario 1, the fantasy goal is to dock safely in time. For scenario 2, the fantasy goal is to avoid a collision.

Is the challenge level appropriate?
Malone mentions that a game should be playable at multiple levels of difficulty, so that it can provide a challenge for players over a range of ability levels. The heuristics include four ways to achieve this.

(a) Variable difficulty level: The difficulty level is determined by the researcher creating the scenario. The player has no way of selecting/changing the difficulty level before or during play.

(b) Multiple level goals: This is somewhat limited, as the player is given only one overarching goal. However, scenario 1 involves a speeded response (dock as quickly as possible), which is a type of meta-goal[9]. Scenario 2 has a partial win state, as previously mentioned.

(c) Hidden information: By not revealing the specific win/fail states of each scenario, but instead giving an overarching goal, it is not certain when (or if) manual takeover is required. This also increases challenge, as the player has to be alert and determine by themselves when they should take control.

(d) Randomness: Randomness is generally not desired for scenarios used in research, and was therefore not used in any way. Because of this, the takeover events both scenarios were fully scripted (except for changes due to user input).

To summarize, the scenarios provide multiple level goals, and use hidden information to create an uncertain outcome. However, both scenarios have only one takeover event, and no way for the player to vary the difficulty level. Therefore, it is likely that the test participants did not perceive difficulty in the same way.
This coincides with the findings: Q9 shows that there is a difference in perceived difficulty, where 3 said 'not challenging', but 2 said 'challenging/hard'. Additionally, 9 said 'boring', while 6 said either 'fun' and/or 'interesting'. In Q10, 'easy' and 'stressful' were both said 4 times.

6.5 Fantasy

While playing a scenario, the player assumes the role of a SCC operator, monitoring an autonomous passenger ferry in-action. In this way, the participant enters a game fantasy by assuming a role different from themselves. The fantasy world is closely related to reality, as it features the Milliamperé 2 ferry (currently under construction), and a real-life location (Trondheim city).

How simulation fidelity helps to increase fantasy
As simulation fidelity essentially defines a simulation's reality[3], it can be used to explore how well the player is immersed in the game fantasy.
As mentioned, the test location itself resembles a shore control center. In this way, the fantasy of being an SCC operator was not limited to the screen only, but was also connected to the location where the simulator was played.

Is the fantasy intrinsic?
Malone mentions that games benefit from having an intrinsic fantasy, as the required skill is often aimed at a real world goal. He also mentions that simulations are examples of this[9]. "In general, intrinsic fantasies are both more interesting and more instructional than extrinsic fantasies."(ibid.)

As previously mentioned, the goals are presented in the context of the game's fantasy (i.e. fantasy goals), which is another quality of intrinsic fantasies [9].

Does the fantasy have an emotional impact?
Q9 shows that 9 found it boring, and presumably had a low emotional response.
However, 1 said it was 'important/meaningful' in Q9, and 2 said they felt responsibility in Q10.
Finally, 3 and 5 said 'fun' in Q9 and Q10 respectively.
This coincides with the difference in perceived difficulty mentioned earlier.

6.6 Curiosity

Sensory curiosity
The environment fidelity mentioned previously works as decorative visual stimuli, and combined with simulated hydrodynamics, it also enhances the fantasy. However, auditory stimuli is more limited, as it mostly consists of ocean ambience and the sound of the ferry's motors.

Sound and graphics can also be used as a representation system to convey information more efficiently than with plain text[9]. This is reflected in the findings of Q14 and Q15, where multiple participants mentioned that the grey and repetitive notifications were hard to read, and should be replaced with either color coding, or iconography.

Cognitive curiosity
Malone present that surprising feedback helps to engage curiosity. The fact that the exact win/fail-

ure states are hidden, provides a potentially surprising feedback once the actual takeover event occurs.
The exact behaviour of surrounding boats, and how the scenario will unfold, is also unknown to the player, which also has the potential to provoke curiosity.

Informative feedback: Feedback when a failure state occur is immediate, and is naturally a part of the simulator's fantasy. For example, when a collision occurs, feedback is provided through audio and how the ferry moves after impact.
In some cases, feedback was given before a failure state, through either an early warning message, or the takeover warning with countdown.

6.7 Further Development

Operator interface and HUD
Q3 in findings clearly show that the test participants found the boat traffic map to be most useful. Therefore, further development should focus on exploring how this map can be used to further increase situation awareness.
As discussed, it would be beneficial to represent different ferry states using color coding and iconography, instead of plain text. The following ferry states would benefit from this:

- Whether the ferry is currently boarding passengers
- Whether the ferry is currently docked
- Whether docking can be performed (useful during manual control)
- Colored proximity warning, either overlayed onto the camera view, or on the map (as suggested in Q14 and Q15).

Other relevant suggestions from Q14 and Q15 include:

- Highlighting the currently selected ferry, and show the camera's field of view cone
- Showing the ferry's planned path on either the map or the camera screen.
- Having a single camera, as multiple participants suggested either increasing field of view, or removing the camera's rotation constraints.

Autonomous operation As mentioned in the Development chapter, a simplified version of the ferry's autonomous systems was implemented. This was done through a combination of scripted scenarios, and a simplified collision avoidance system.
While this scripted approach was useful for the user testing purposes of Case 1 and Case 2, it is not applicable for researchers interested in detailed testing of the autonomous systems found on the actual Milliamperé 2 ferry. To this end, it is relevant to explore whether the existing Gemini project[1] can be integrated into the simulator. Gemini's detailed simulation of Milliamperé's sensors is a great resource for more closely simulating the ferry's situational awareness and collision avoidance capabilities[2].

Maintaining the operator's attention
The discussed issue of boredom and inattention is an important one in the context of the SCC operator role. Further development should focus on how to either retain an operator's attention, or how to quickly and consistently grab their attention when required.
Further exploring the use audio as informative feedback is highly relevant here, as it does not require the operator to look at the screens.

6.8 Contributions

Perhaps the main contribution of this master's project, is the development of a specific tool that was designed based on feature requests from researchers at SCL, and that was used as a central component in their experiments. This main contribution can be split into three main parts: (1) the scenario builder for creating and changing scenarios, (2) the simulator for user testing, and (3) automatic data collection while a scenario is played.

Through the scenarios used in SCL's experiments, two specific tail events were tested, namely safely handling a system failure, and avoiding a potential collision. Another contribution is simulator's automatic data collection

Multiple existing design prototypes from previous or ongoing research projects were implemented and integrated into the simulator:

- The scenario builder, designed and developed through a collaboration between the author and Sondre Ek [6]
- The heads-up display for ferry monitoring and operation, designed by Erik Scott Hognestad [4]
- The design concept for a sea map and ferry overview screen, providing additional situational awareness. Designed by Jooyoung Park [11]

6.9 Limitations

- Because the development essentially was split up into two parts, it became difficult to give them the same amount of attention throughout the project (i.e. the simulator and the scenario builder).
- It would be beneficial to have more rigorous and standardized user tests regarding the scenario builder, but due to time constraints, the focus was instead moved towards the experiments and creating the required scenarios.
  - For example, it would be beneficial with interview questions regarding simulation fidelity, use of auditory/visual feedback (within the simulated environment), etc.
- All researchers who used the simulator were in some way part of the same team and/or experiment. This was a likely outcome when deciding to create the simulator for the SCL team's needs specifically, and targeting a specific experiment. However, a more thorough approach would likely be too time consuming and too large in scope for this master project, and seem to be a necessary limitation.
- The simulator was developed with the assumption that it should be used in the same environment throughout (i.e. the SCL), with a specific monitor setup and hardware specifications in mind. If the simulator is to be used in other contexts, is should be made less resource-demanding, and should target other hardware/platforms.
- Keybindings and overall interfacing with the scenario builder should be further user tested, and given redesigns/"quality-of-life" improvements accordingly.
- Any given scenario essentially provide no replayability, as that would mean no uncertainty for the player, meaning that they will quickly lose interest (and will likely produce no usable data). However, as these scenarios are primarily used for research, it is beneficial that a specific scenario always plays out the same way (apart from differences due to user input).

7 Conclusion

In this thesis, the development of a simulator aimed at testing remote operation of autonomous passenger ferries was presented. A scenario builder tool was also developed, so that researchers could create their own scenarios. To ensure that this development was relevant, a case study targeting relevant ongoing research was performed. Findings were presented and then discussed, using Malone's heuristics for designing instructional computer games as a framework. Finally, changes for further development were proposed.

The research question was:

> Can video game technology be used to enhance early-stage research and prototyping of remote operation systems for unmanned (maritime) vessels?

Our findings indicate that game technology may enhance early-stage research and prototyping in the following ways:

- Development has shown that using a video game engine has accelerated the creation of a simulator prototype, and the prototyping process. This allowed the simulator to have a high-fidelity environment that closely represented what an autonomous ferry in action might look like in a real-life scenario.
- The simulator also helped to collect data regarding tail events that would be impractical to test in the real world.
- The inclusion of a scenario builder also helped with quickly creating scripted scenarios, without having to consider technical details.

However, one should apply game design principles with caution, as focusing too much on creating an engaging game may prove detrimental both in terms of simulation fidelity, and for research purposes. While there are clear overlaps between creating an immersive game fantasy and providing a convincing simulated environment, one must ensure that the game aspect does not negatively affect the accuracy of the simulation.

References

[1] "T. Skarshaug" "K. Vasstein". *Gemini Simulator*. URL: https://github.com/Gemini-team/Gemini.

[2] Edmund F Brekke et al. 'milliAmpere: An Autonomous Ferry Prototype'. In: ().

[3] Yngve Dahl, Ole A Alsos and Dag Svanæs. 'Fidelity considerations for simulation-based usability assessments of mobile ICT for hospitals'. In: *Intl. Journal of Human–Computer Interaction* 26.5 (2010), pp. 445–476.

[4] *Design of a GUI for autonomous ferry remote control*. URL: https://www.ntnu.edu/web/shorecontrol/news/-/blogs/design-of-a-gui-for-autonomous-ferry-remote-control.

[5] UNITED NATIONS DEPARTMENT FOR ECONOMIC and SOCIAL AFFAIRS. *SUSTAINABLE TRANSPORT, SUSTAINABLE DEVELOPMENT: Interagency Report/ Second Global Sustainable... Transport Conference*. UN, 2021.

[6] Sondre Ek. 'Design of simulator for researching autonomous marine vessels'. MA thesis. NTNU, 2022.

[7] *https://en.wikipedia.org/wiki/Arrow*$_k$*eys#WASD*$_k$*eys*. URL: https://en.wikipedia.org/wiki/Arrow_keys#WASD_keys.

[8] *IMO takes first steps to address autonomous ships*. URL: https://www.imo.org/en/MediaCentre/PressBriefings/Pages/08-MSC-99-MASS-scoping.aspx.

[9] Thomas W Malone. 'What makes things fun to learn? Heuristics for designing instructional computer games'. In: *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*. 1980, pp. 162–169.

[10] *NTNU Shore Control Lab*. URL: https://www.ntnu.edu/shorecontrol.

[11] Jooyoung Park. 'Interface design for a remote operator to monitor and control multiple autonomous ferries'. MA thesis. NTNU, 2022.

[12] Erik Aleksander Veitch, Thomas Kaland and Ole Andreas Alsos. 'Design for resilient human-system interaction in autonomy: the case of a shore control centre for unmanned ships'. In: *Proceedings of the Design Society* 1 (2021), pp. 1023–1032.