# EcoQuest

University of Southern Denmark

Faculty of Engineering(TEK)

Software Engineering

## Project duration:

September to 2nd January, 4 months

## Semester:

1st semester

## Course code:

SI1S-PRO

## Project Group:

Group 20

## Project authors:

Victor Petrica vpetr24@student.sdu.dk,

Toni Vera Gutierrez tover24@student.sdu.dk,

Eduard Gangan edgan24@student.sdu.dk,

Gene Enrick Miguel Giroy gegir24@student.sdu.dk,

Deividas Petrulis depet24@student.sdu.dk,

Matus Ilavsky milav24@student.sdu.dk

## Project supervisor:

Dominik Wojciech Bosy dobos23@student.sdu.dk

# Table of Contents

# 1.3 Abstract

EcoQuest is a text-based educational game created to raise awareness about marine conservation, supporting the United Nations Sustainable Development Goal (SDG) #14: Life Below Water. The game immerses players in environmental challenges such as pollution, overfishing and habitat destruction. Players take on the role of a marine biologist and activist, making strategic decisions to protect marine ecosystems while learning about these real-world scenarios. The game encourages players to consider their environmental impact, providing an interactive experience that highlights the importance of sustainable practices.

Inspired by World of Zuul and developed using the C# programming language with the .NET framework, EcoQuest features diverse underwater and coastal settings. Using text commands players navigate these environments, interact with non-player characters, and engage in sustainable practices such as removing plastic waste or supporting marine protected areas. The game's educational content is presented through dynamic storytelling and challenges, making learning about marine conservation engaging and immersive. A personalized impact report at the game's conclusion summarizes the player's contributions, reinforcing the importance of sustainable actions.

EcoQuest includes a reputation system that rewards players for making environmentally friendly choices, allowing them to unlock new locations and meet different characters. The game shows how player decisions affect the world, helping them understand the importance of making good choices. Changes in the environment, such as cleaner beaches and healthier marine life, make the experience more engaging and rewarding. Players can see the consequences of their actions, which reinforces the connection between their in-game choices and real-world environmental issues.

By combining fun gameplay with educational lessons, EcoQuest helps players learn about the problems facing our oceans and think about ways to protect them. The game demonstrates how video games can be used to teach important topics like marine conservation, making the learning experience both enjoyable and impactful.

# 1.4 Introduction

EcoQuest is a game created to promote awareness about protecting life below water turning education into a hands-on activity. In this game, participants become environmental champions responsible for safeguarding different marine environments against dangers like pollution, overfishing, climate change, and habitat destruction. Throughout the journey, players are informed about the environmental consequences of these problems and the steps they can take to improve the health of the ocean. In this way, EcoQuest supports the United Nations Sustainable Development Goal (SDG) #14: Life Below Water that emphasizes the conservation and sustainable utilization of oceans, seas, and marine resources for sustainable development.

## Problem Statement

Marine life is degrading due to human activities and there is still a significant scarcity in educational material through gamification to promote awareness of the issue. How can a text-based game create awareness amongst its players on the consequences of human activities on marine life?

# Problem Analysis

### General Description of the Problem

The degradation of marine ecosystems has become one of the most pressing environmental challenges of our time. Human activities such as pollution, overfishing, plastic waste accumulation, and coastal development have led to severe consequences, including the destruction of coral reefs, declining fish populations, and the loss of biodiversity. These issues not only threaten the health of marine life but also have far-reaching implications for global food security, climate regulation, and the livelihoods of millions who depend on oceans for their sustenance and income.

Despite global initiatives like the United Nations' Sustainable Development Goal (SDG) 14: "Life Below Water," which focuses on conserving and sustainably using oceans, seas, and marine resources, progress remains slow. This is partly due to insufficient public awareness of the interconnectedness between human actions and their environmental impacts. Without a deeper understanding of these issues, many individuals lack the motivation to adopt sustainable practices or advocate for necessary policy changes. Addressing this gap in awareness and fostering a culture of environmental responsibility are critical to preserving marine ecosystems for future generations.

### Specific Description of the Problem

A key challenge in promoting marine conservation is the limited effectiveness of traditional educational and awareness campaigns. These approaches often rely on passive methods, such as lectures, pamphlets, or videos, which fail to actively engage audiences, especially younger generations who are more accustomed to dynamic and interactive media.

Interactive platforms, such as video games, offer a promising solution to this challenge by providing immersive experiences where players can learn through direct involvement. However, designing games that balance entertainment with meaningful educational content is a complex task. Such games must ensure that players not only enjoy the experience but also gain a deeper understanding of the consequences of their virtual actions and how they relate to real-world environmental challenges.

For instance, while many games include themes of exploration or resource management, few explicitly connect these mechanics to sustainable development principles. Additionally, incorporating systems that provide real-time feedback and demonstrate the cause-and-effect relationships between actions and outcomes is crucial for creating a lasting impact on players' attitudes and behaviors.

## 1.5 Method and Project Delimitation

The primary goal is to educate players about ocean conservation, focusing on real-world problems such as pollution, overfishing, climate change, and habitat destruction. Through interactive storytelling, players learn the importance of sustainable practices and how human activities affect marine life.

The game is developed using the C# programming language with the .NET framework, providing a smooth and responsive gameplay experience. Players explore various marine ecosystems, such as coral reefs, seagrass meadows, and kelp forests, all of which are threatened by environmental issues. As they progress, they are faced with complex decisions on how to tackle these problems, whether by cleaning up plastic waste or establishing marine protected areas to preserve biodiversity.

Every decision has consequences, with the game simulating the long-term effects of the players' actions. For example, the decision of the player ignoring pollution could lead to the degradation of the marine ecosystem. This approach encourages players to reflect on the impact of their choices and understand the importance of responsible actions for ocean conservation.

One of the standout features of EcoQuest is its environmental education. As players make decisions, they receive detailed explanations of the environmental issues they encounter and how their choices affect the ecosystem. This includes not just cleaning up waste or protecting species, but also understanding complex phenomena like ocean acidification and coral bleaching due to climate change.

## 1.6 Requirements

**<u>Functional Requirements</u>**

The game is built around user-controlled text input, allowing players to interact with the environment, NPCs, and objects through typed commands. The following functionalities are essential:

- **Player Navigation:**
  - Players can move between rooms or locations using directional commands (e.g. “west”, “north”, “south”, “east”, “sail”).
- **Reputation System:**
  - Players gain reputation by performing environmentally beneficial actions such as recycling trash.
  - Reputation unlocks NPC interactions, new locations, and game upgrades.
- **Object Interaction:**
  - Players can interact with objects(trash), such as picking them up or sorting them for recycling.
- **Dynamic Environment:**
  - Room descriptions change based on player actions, reflecting environmental improvements (e.g., a polluted beach becomes clean, recycling station is now functional).
- **NPC Interaction:**
  - Players can initiate dialogues with NPCs, triggering quests or unlocking new features.
  - NPCs also have the role of informing and educating the player on specific environmental issues through their dialogue.
- **Command System:**
  - Basic commands (e.g., “sail,” “look,” “sort”) control the player's actions, ensuring simplicity and usability.

- **Interactive Minigames:**
    - 3 main Minigames implemented in the game to create an interactive experience:
        - Trash Sorting Minigame: Players sort the collected trash from the beach.
        - Fish Tagging Minigame: Players find and select endangered species of marine life which need to be tracked and protected.
        - Quiz Minigame: Players are tested on their knowledge about problems encountered by the marine environment.
- **Educational Feedback:**
    - Players will receive immediate feedback, either with the correct answer(Recycling Minigame) or an opportunity to correct themselves(the player can go to the library to find out the right answers for the Quiz Minigame) , reinforcing learning and enhancing their understanding of the game's content.
- **Environmental Consequences:**
    - Players' actions directly impact the game world, demonstrating environmental cause-and-effect relationships.
- **Linear progression:**
    - The player should move to another location once he solves the environmental issues in the current location.
- **Winning condition:**
    - The game is finished once the player has solved the environmental challenges in all locations and is tested in a quiz minigame to ensure it was an educational experience.

## Non-Functional Requirements

- **Performance:**
    - The game should load quickly and remain responsive on entry-level devices.
- **Reliability:**
    - Commands should execute without errors, and the system should handle exceptions gracefully.

- **Ease of Use:**
    - The game interface must prioritize simplicity, ensuring smooth gameplay for all users.

## User Requirements

- **Playability:**
    - The game must be intuitive, with clear instructions for navigation and interactions.
- **Accessibility:**
    - Text and controls should cater to new players, emphasizing inclusivity.
- **Educational Value:**
    - The game should teach players the value of environmental stewardship through engaging gameplay mechanics.

## System Requirements

- **Compatibility:**
    - The game should function on basic desktop systems without requiring advanced hardware.
    - The game should be cross-platform ensuring that it works on most operating systems.
- **Development Environment:**
    - The game is written in C# using the .NET framework, ensuring compatibility with modern development tools.

## Use Cases and User Stories

- **Movement:**
    - *As a player,* I want to navigate between rooms using simple commands so that I can explore new areas.
- **Item Interaction:**
    - *As a player,* I want to pick up and recycle trash found in rooms to improve the environment and gain rewards.

- **NPC Dialogue:**
    - *As a player,* I want to engage in dialogues with NPCs to uncover quests and unlock new areas or features.
- **Dynamic Environment:**
    - *As a player,* I want to see the environment change based on my actions, giving me feedback on my progress.

**<u>Noun-verb analysis</u>**

| Nouns | Verbs |
|---|---|
| Game<br>Text-based<br>User-controlled input<br>Commands<br>Text description<br>Map<br>Room<br>Trash<br>NPC<br>Dialogue<br>Quest<br>Description<br>Environment | Is<br>Depends on<br>Can<br>Cause<br>Look up<br>Navigate<br>Changes<br>Recycle<br>Starts<br>Triggers<br>Engages<br>Updates<br>Reflects |

**CRC Card for NPC Class**

**Class name: NPC**
**Responsibilities:**
-Initiate Dialogue with the player

-Cause consequences on the environment

-Start quests

-Start minigame

**Collaborations:**

-Room(where NPCs are placed in, NPCs can change room description)

-Quest(to start)

-Quiz Minigame(to start)

-Recycling Station(repair it)

-Upgrade System(Mayor NPC opens the Upgrade Menu)

-Inventory(upgrade inventory space)

-Dialogue(NPCs have a Main Dialogue that starts when the player talks to them)

## 1.7 Design

The design of *EcoQuest* is structured to achieve its objective of spreading awareness by educating players about marine conservation and delivering an engaging, interactive gameplay experience. This section outlines the system's structure, organization, and implementation strategies to ensure that the game aligns with its educational and interactive goals.

## Educational Purpose and Theme

*EcoQuest* serves as an interactive text-based platform to educate players about the conservation of our oceans. It draws inspiration from the United Nations' Sustainable Development Goal 14: *Life Below Water*, which focuses on preserving marine ecosystems. By introducing players to real-world environmental challenges such as pollution, overfishing, habitat destruction, and climate change, the game aims to raise awareness

## Core Gameplay Design

**1. Storyline**

To enhance the learning experience, *EcoQuest*'s environment has a dynamic storyline that evolves alongside the player's progress. The environment adapts to the player's actions, offering new mechanics and context as they advance. This progression gives a deeper understanding of marine conservation topics, making the learning experience both immersive and meaningful.

**2. Reputation System**

The reputation system provides players with a sense of responsibility by rewarding ethical decision-making and sustainable choices. It encourages critical thinking about conservation strategies and affects gameplay outcomes by influencing NPC interactions, gameplay, privileges, and access to certain quests and resources. This system ensures the player feels the impact of their choices, enhancing both engagement and educational value.

**3. Quests**

Quests in *EcoQuest* are designed to address real-world issues to the player, providing hands-on learning opportunities that help educate and engage. Each quest tackles challenges such as cleaning polluted beaches or addressing overfishing. Quests are structured to promote critical thinking, problem-solving, and a sense of responsibility for sustainable actions. By completing quests, players progress in the game but also absorb valuable lessons aligned with SDG 14.

**4. Mini-Games**

Mini-games in *EcoQuest* are made to provide interactive ways for players to engage with conservation challenges, combining education with entertainment. Mini-games in *EcoQuest* include sorting trash at recycling stations and fish sorting for tagging endangered marine species due to overfishing. These mini-games offer hands-on learning experiences, encouraging players to connect gameplay actions with real-world environmental responsibilities, enhancing both their skills and understanding of marine conservation aligned with SDG 14.

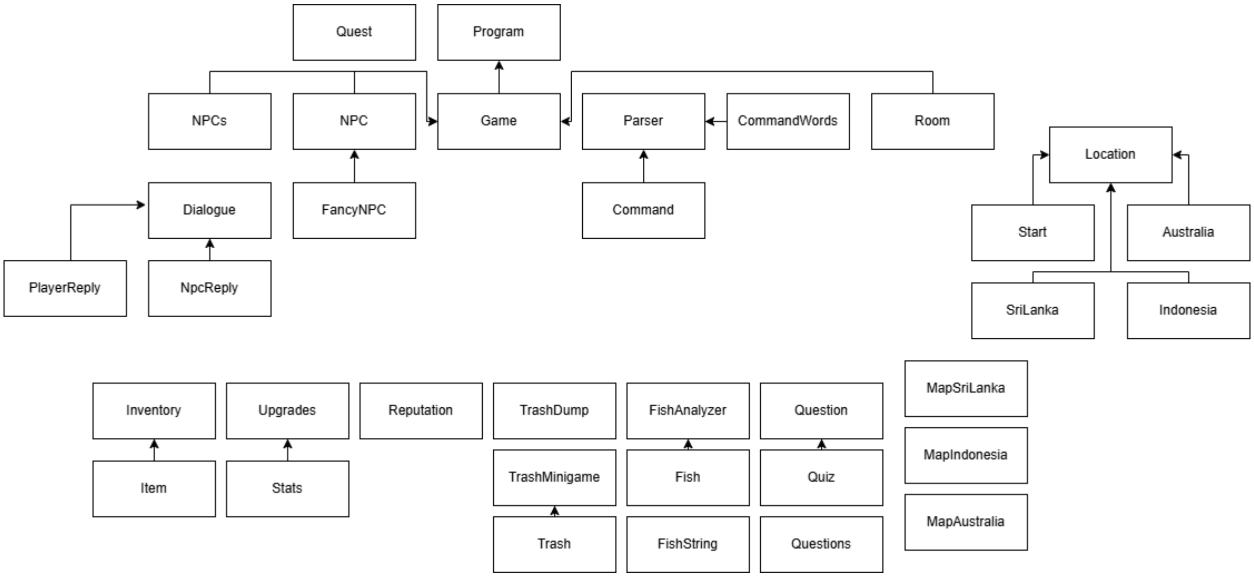
## System Architecture

The system architecture of EcoQuest follows a modular design approach, ensuring the game's structure is scalable, maintainable, and flexible for future expansions. This approach breaks down the game's code into independently manageable and reusable components, each responsible for a specific aspect of the game's functionality. This design ensures that each part of the system can be developed without disrupting the other parts, which makes the game easier to maintain and expand.

## User Interface Design

The user interface design for EcoQuest is currently displayed through VS code's terminal, relying entirely on text-based commands and outputs. User interface also includes the use of Spectre.Console. Spectre.Console provides visual appeal through features like colored text, dynamic text effect, table layout and visual image, that creates a more engaging experience than traditional plain text.

## UML Diagram



Simplified Version

## Performance and Scalability

The code for the system is designed to be simpler and independent in order to handle future expansions, including new quests, mini-games, and more complex mechanics.

## Algorithms and Logic

Algorithms in EcoQuest are designed through the player's decisions, determining whether the outcome gives a positive or negative effect. Every action that the player takes, whether it is completing a quest or a mini game, it affects the game's environment and the player's reputation.

**Positive outcome**, when the player successfully sorts trash in the sorting mini game or completing a quest, this results in an increase in reputation or gain something from NPC.

**Negative outcome**, when the player fails sorting trash in the sorting mini game or ignores a quest, this will result in a decrease in their reputation.

# 1.8 Implementation and testing

## Development technologies

For the development of ***EcoQuest***, we used the programming language **C#** because it was the most familiar language for all group members. We used **Visual Studio Code** as our code editor and **Git** as our cloud storage as a solution for the best cooperation within the team. For task distribution, we used **Trello**, and for communication, we relied on the popular platform **Discord**.

The base game for ***EcoQuest*** was a template game called *Word of Zuul*, meaning players might notice some similarities in structure at certain moments. However, *EcoQuest* is a completely different game. Since we developed the game in **C#**, our team decided to use available libraries. For the visual aspect, we used the **Spectre Console** library, which made the game more engaging and intuitive for players compared to earlier versions.

As mentioned, we used **Git** for cloud storage, which worked perfectly in cooperation with Visual Studio Code for team collaboration. To minimize bugs, we followed a strict rule: no one was allowed to push non working code. This prevented potential disruptions for other team members, who might otherwise have been unable to run the project and test their parts. Additionally, we implemented several internal rules to make the development process as smooth as possible.

**Discord** also served as a centralized hub for all necessary information and links, such as resources for creating presentations on Canva. At each in person meeting, we tested the current version of the game, looked for improvements and also in some cases for bugs.

For report writing, we chose **Google Docs** as the best option, in our opinion, because it allows for a shared document where all members could contribute by writing their parts.

During the process, we primarily used **Notion** for making notes, but we were forced to switch after realizing that the free version was insufficient for the scope of the project we were planning.

## Coding conventions

We also stuck to basic coding conventions. These included:

**Readability**: Ensuring that every team member could navigate the code without issues.

**Maintainability**: Making the code easy to update and modify as we frequently changed parts during development.

**Efficiency**: Optimizing the code to avoid unnecessary hardware usage.

**Reusability**: Writing concise and modular code, often leveraging inheritance, such as in the location and NPC classes.

## Tasks distribution

Task distribution was managed via **Trello**, where each team member could view their assignments for the exact week. Tasks were assigned during weekly in person meetings, where we discussed the next steps. Depending on the difficulty, members were given one or more tasks to complete by the next meeting. If someone had some issues, they could contact the rest of the team through Discord for help. In such cases, we used **Discord** for online meetings.

## Testing

For testing, we have used two different approaches: **black box** testing and white box testing. **White box** testing was conducted by team members who had knowledge of the functionality, while black box testing was carried out by individuals without prior knowledge of the system's inner workings. The focus of black box testing was to identify bugs and improve the overall functionality of the game. On the other hand, white box testing concentrated on enhancing the game's design and making it more engaging and entertaining for players.

## Development process

Our development process began with brainstorming ideas for what we could create. After selecting our topic, we merged with another group, and the development of the game gradually started. For development, we used all the tools mentioned earlier. Leading up to the first presentation to the examiners, we focused on developing the functionality and gameplay. However, after the first presentation, our focus shifted to the visual aspects. At this stage, we mainly used Spectre.Console to enhance the game's appearance and make it more engaging for the player, aiming to hold their attention for a longer period. Spectre.Console offered us many features, from making text appearance more interesting to designing menus and incorporating pixel art. Additionally, our program included ASCII art, particularly in the Indonesia section. For example, when navigating the submarine, instead of a standard map, players would see an ASCII art representation. Throughout the programming process, we also worked on writing the report and creating a UML diagram. Once the programming phase was complete, the entire team began testing the game over a period of time to identify and fix as many bugs as possible before submitting the project. Finally, the last step was to complete the report, which marked the final milestone in our project journey.

## 1.9 Evaluation

- **Assessment of Requirement Fulfillment:**

To evaluate whether the solution meets the project's requirements, user testing was conducted, and it was compared directly to the criteria defined at the start. Overall, users mentioned that the solution fulfills the main functions and provides an intuitive experience, aligning with the objectives set for the project. While some minor details were identified that could be improved, the solution largely meets expectations. In summary, the project fulfills the main requirements, but there is room for adjustments to make it even better.

- **Achievements vs Expectations**

The project achieved several key turning points aligned with the initial objectives. These successes demonstrate the project's ability to deliver on its primary goals.

- **Strengths and Weaknesses**

One of our strengths during this project was without any doubt our delightful communication skills, this helped us to understand each other's points of view and made it easier to work. Furthermore, it also helped the fact of being very constant and serious with the work, attending the meetings (except those that due to force majeure could not be attended), this has helped us to follow a well-matched work plan and reach the date with plenty of time.

On the other hand, something we could improve is the way we present the results visually. Although the graphs and tables used were clear, exploring more advanced tools or techniques could have made the presentation even more impactful and professional.

- **Potential for better results**

As previously mentioned, an improvement that we could have implemented would be in the visual aspect of the project. That said, there is always room for improvement in any project, but we believe that our game is efficient and adequately responds to the objectives set. We are satisfied with the decisions made throughout development, as they reflect an appropriate balance between quality and functionality.

- **Known issues**

Throughout the project, we encountered various challenges, ranging from minor programming errors to issues in the report. One of the most significant difficulties was incorporating images into our work in a way that maintained their visual quality on the screen. Resolving this issue required several hours of experimentation and collaborative effort. Ultimately, through teamwork and persistence, we successfully identified a solution. This experience highlighted the importance of problem-solving and collective determination in overcoming obstacles.

## 1.10 Conclusion

The main objective of the EcoQuest project, which was to develop an instructional text-based game that encourages sustainable habits and increases knowledge of marine conservation, was accomplished. The game addressed the important environmental issues mentioned in the problem statement, like pollution, overfishing, and habitat degradation, by skillfully fusing entertaining gameplay with instructional material. The research showed how interactive media may successfully communicate difficult environmental topics and encourage sustainable practices by putting players in the shoes of marine biologists. The creation of a dynamic reputation system to encourage ecologically conscious choices and the addition of real-time feedback mechanisms to improve players' comprehension of the effects of their activities are two notable accomplishments, creation of a modular game architecture for scalability, and implementation of engaging mechanics like quests and mini-games that align with real-world conservation challenges. Despite challenges such as balancing educational content with entertainment and presenting results effectively, strong team communication and commitment ensured success. EcoQuest answers the research questions posed in the problem statement by demonstrating how interactive media can foster awareness and inspire action for marine conservation, providing an innovative approach to environmental education that combines learning with entertainment.

# 1.11 Process Evaluation

## Learning Process:

The development of this project has been an enriching experience for the team, fostering both personal and professional growth. Throughout the journey, the team has acquired a wide range of skills, including problem identification, solution development, and the implementation of these solutions. All of this was achieved while working collaboratively, which proved to be a pivotal aspect of the project.

Teamwork played a crucial role in the development process. For many members, this project was their first opportunity to work in a structured team environment. This experience has been transformative and highly educational, teaching us valuable lessons about communication, collaboration, and conflict resolution. We learned how to leverage the strengths of individual team members and address challenges collectively, creating a supportive and efficient working dynamic.

On the technical side, some team members began the project with little to no knowledge of Object-Oriented Programming (OOP). However, through self-motivation and the guidance of more experienced team members, they were able to develop meaningful programming skills. This collaborative learning environment not only enhanced individual competencies but also reinforced the team's ability to share knowledge and grow together.

## Team Roles and Dynamics:

The team operated without formally designated roles, with the exception of the Team Leader. The Team Leader served as the primary point of contact with the Project Supervisor and was responsible for organizing meetings. This flexible structure allowed all team members to contribute across various areas of the project without constraints. The lack of rigid role assignments fostered a collaborative environment where individuals could leverage their strengths and engage in diverse aspects of the project. This approach significantly enhanced team dynamics and played a crucial role in the success of the project.

## Cooperation:

The group demonstrated strong internal collaboration, working together seamlessly with minimal issues. Decisions were made collectively through a voting process based on majority rule, ensuring that all members had an opportunity to contribute. Designated meetings provided a platform for team members to suggest ideas and participate actively in the game design process. When uncertainties arose, the team maintained open and effective communication with the Project Supervisor. Their guidance was instrumental in addressing challenges and resolving any issues encountered during the development process. This combination of internal cooperation and external support greatly contributed to the project's overall success.

## Work Methodologies:

During the project execution phase, the team adopted a structured approach to ensure effective collaboration and alignment with project goals. Weekly meetings were held, either in person or online via Discord, to discuss the next steps and outline tasks to be completed before the following meeting. These tasks were organized and tracked using Trello, which facilitated efficient distribution of responsibilities among team members. To prepare for important deadlines, additional meetings were scheduled to review progress and ensure the quality and completeness of the work. This iterative approach allowed the team to address any issues proactively, ensuring that all tasks were aligned with the project's objectives. The combination of regular check-ins, task management tools, and adaptive scheduling proved highly effective in achieving project milestones and maintaining overall project cohesion.

## Writing Process:

The Report writing process was a collaborative effort, with all group members contributing equally. Sections were divided among members to ensure a balanced workload, and written material was reviewed by others to enhance clarity and correctness. This iterative review process not only improved the quality of the content but also ensured that the final document was cohesive and aligned with the project's objectives.

## Time Management:

The project's time management was highly effective, as all major submission deadlines were successfully met. Internal weekly deadlines were also consistently achieved, owing to efficient work distribution and strong collaboration among team members. This proactive approach ensured steady progress and allowed the team to address tasks systematically.

## Leadership:

The group benefited from having a designated leader who played a pivotal role in guiding the project. The leader effectively communicated with the group supervisor, organized meetings, facilitated work distribution, and coordinated the creative process. This leadership approach ensured that the team remained focused and cohesive, contributing significantly to the project's overall success.

## Work Distribution:

Project tasks were assigned during weekly meetings, following a structured process. Initially, tasks were identified and outlined, after which team members selected tasks that aligned with their strengths and interests. Trello was utilized to facilitate task distribution and set deadlines, ensuring transparency and accountability. The group leader played a crucial role in maintaining fairness and effectiveness by overseeing the distribution process and ensuring that workload was balanced among team members. This approach fostered a sense of ownership and collaboration, contributing to the project's smooth execution

# Appendix I

## Collaboration Agreement

**Date**: 25th Semester 2024

**Project Title:** Reducing human impact in order to preserve life below water (SDG#14)

**Group Number:** 20

**Group Members**:

- Victor Petrica,
- Toni Vera Gutierrez,
- Eduard Gangan,
- Gene Enrick Miguel Giroy,
- Deividas Petrulis,
- Matus Ilavsky

As a group, we agree to work on the mentioned project title.

We will follow the project guidelines and descriptions.

We will maintain open and respectful communication within our group. In the event of any disagreements, we will engage in constructive discussions and work toward mutually acceptable compromises.

We will respond to emails and messages promptly.

We will schedule regular meetings and ensure that all members attend and are informed in advance, presence at meetings is mandatory, if one can not attend the meetings they should inform the group beforehand.

We agree to contribute equally on coding the project and writing the report. We should not do the project individually so everyone can participate and learn.

Important decisions will be made according to majority rule.

We agree to abide by the terms and guidelines outlined in this group agreement and project guidelines and descriptions.

Other conditions may be added eventually if the majority decides so.

# Appendix II

## Supervisor Agreement

**Date**: 25th Semester 2024

**Project Title:** Reducing human impact in order to preserve life below water (SDG#14)

**Group Number:** 20

**Group Members**:

- Victor Petrica,
- Toni Vera Gutierrez,
- Eduard Gangan,
- Gene Enrick Miguel Giroy,
- Deividas Petrulis,
- Matus Ilavsky

We will listen carefully to the Supervisor’s advice and suggestions, recognizing their expertise and experience in guiding the project.

We will respect the Supervisor’s role by addressing concerns professionally and treating all feedback with due consideration.

We will keep the Supervisor regularly informed of project progress, challenges, and any significant changes.

We agree to abide by the terms and guidelines outlined in this group agreement and project guidelines and descriptions.

We ask of the supervisor that they respond to messages, questions promptly.

This agreement may be modified in the case that both the group and the supervisor agree to do so.

This agreement between **Dominik Wojciech Bosy** and **Group 20** is valid from 25th September 2024 until the successful completion of the project.

# Appendix III

## Project Log

**LINK TO PROJECT LOG**

# Appendix IV

## Portfolio Checklist

| Chapter | Content | Fulfilled +/- |
|---|---|---|
| Cover page | Project title, educational institution, faculty, department, program, course code, project period, supervisor, project group, and project participants (first name, last name, SDU email).<br>May include illustrations. | + |
| Table of Contents | Comprehensive table of contents for the entire project portfolio. | + |
| Introduction | Project context and background for the project. | + |
| Problem analysis | Description of the delivered framework.<br>Problem statement and limitations.<br>Methods.<br>Timeline. | + |
| Requirements | Total requirements (received and formulated by you).<br>Analysis of the requirements, including:<br>• Noun<br>• Verb analysis<br>• CRC cards | + |
| Design | Description of the design, including one or more UML class diagrams. The description includes architecture and user interface design. | + |
| Implementation | Description of the implementation with key parts of the program code. | + |
| Evaluation | Evaluation of whether the solution meets the requirements, e.g., through user surveys or user testing.<br>Description of what has been achieved and what has not been achieved in the project compared to the expected outcomes described in the introduction. Description of strengths and weaknesses in the results and whether better results could have been achieved. Description of known issues. | + |
| Iteration #1 and #2 | Have results from both iteration #1 and iteration #2 been included in the above sections (Requirements, Design, Implementation, and Evaluation)? | + |

| | | |
|---|---|---|
| Conclusion | Summary of the results and the evaluation.<br>Answer to the problem statement. | + |
| Process Evaluation | Reflection on the process and the group's consideration of the process: The learning process, team roles, internal group and supervisor cooperation, project work methods, working methods, methodologies, writing process, time management of the project, project leadership, project work distribution, etc. | + |
| Collaboration Agreement | Insert the collaboration agreement as Appendix. | + |
| Supervisor Agreement | Insert the supervisor agreement Appendix. | + |
| Project Log | Address and a link to the project log Appendix. | + |
| Portfolio Checklist | Completed Portfolio Checklist Appendix. | + |