

Workshop 3 - Requirements, Specifications, and Design

Due

Thursday, February 27th at 11:59 pm.

Instructions

Your writeup, either typed or handwritten, should be a single PDF file. For requirements and specifications, feel free to use \LaTeX , Word, or handwritten. For your UML class diagram, you must use Mermaid to make your UML Class design, save the designs as images, then input those images into the final PDF submission file. This submission is on Canvas. This is an individual submission. Every student in the course must get hands on experience with this workshop.

Problem - Aquarium Simulation System

We work for a company that specializes in creating simulation systems. Our latest customer has ordered an Aquarium Simulation System. We will be designing the simulation using a fast version of the Waterfall model. We will stop right after the design phase (no coding and no testing).

Sketch

First, sketch the system and list the components (actors).

Literally draw a diagram of an aquarium. EARS style requirements fix some issues about writing requirements, but they don't solve omission.

Sketching a diagram (literally drawing it out) will help you find and discover actors in the system that you may have forgotten about if you hadn't done this step.

When you sketch out the system, you will most likely have less omissions than if you didn't sketch the system.

Organizing System into Subcomponents

Break the system into subcomponents. For example, you might have a climate control subsystem or a cleaning subsystem. Once you have all your subcomponents, list them in numerical order. For example: "1. Climate Control, 2. Fish Behavior System, 3. ..."

Requirements

We will be writing 12 functional requirements for the system. You must have at least 2 of each type of requirement from EARS. (This includes 2 complex requirements.) Additionally, after the requirement, state which type of requirement it is. For your convenience, here are all the types: ubiquitous, event-driven, state-driven, unwanted behaviors, optional features, complex requirements.

Each requirement should follow the formula for EARS-style requirements. For example, a ubiquitous requirement would follow the formula: “The <system> shall <system response>.”

Each requirement should be organized under your numbering system. For example:

- **Requirement 1.1 Climate Control:** The climate control system shall always measure the pH level of the water. (ubiquitous)
- **Requirement 2.1 Fish Behavior System:** The Fish Behavior System shall continuously update the position of each fish based on its movement patterns and environmental factors. (ubiquitous)
- **Requirement 2.2 Fish Behavior System:** The Fish Behavior System shall allow AI integrated fish behaviors. (optional feature)

Specifications

List your specifications for the list of requirements. All “how” questions should be answered. Each requirement should have at least one specification associated with it. Usually, there are multiple specifications per requirement.

For each specification that correlates to a requirement, use the same numbering system. In the following example, it is the specification for requirement 2.1, so we label it as 2.1.1:

- **Specification 2.1.1 Fish Behavior System:** The Fish Behavior System shall update the position of each fish every simulation tick (e.g. every 0.1 s). Fish movement shall be determined by a combination of user-defined behaviors, species-specific swimming patterns, and environmental factors such as water currents. The update shall be processed using a physics engine that accounts for speed, direction, and collision avoidance.
- **Specification 2.2.1 Behavior Model Architecture:** The AI model shall use a finite state machine (FSM) or behavior tree to manage fish behaviors. Each fish shall have an individual behavior profile based on species, environment, and interaction history.
- **Specification 2.2.2 Species-Specific Behaviors** Fish shall exhibit distinct behaviors based on their species, including: schooling where certain species (e.g., tetras) shall swim in coordinated groups within a proximity

radius (e.g., 10 cm), territoriality where aggressive species (e.g., cichlids) shall defend a defined space (e.g., 20 cm radius) and chase intruders, predation where if predatory fish are included, they shall attempt to chase smaller fish within a certain range, and hiding where shy species (e.g., loaches) shall seek cover near rocks or plants when startled.

Notice that there are multiple specifications for one requirement in the example above. Your specifications don't need to be as specific as these ones, however, they do need to address **how** your requirements will be solved.

Design

Go to Mermaid and make a UML class diagram for your system. Each actor should be represented as a class. Classes do not need any functions or attributes for this assignment. Make the relationships as best as you can though. Refer to the homework 2 write-up for help on the relationships between classes. There are six that would make sense in this context: association, dependency, inheritance, realization, aggregation, and composition. Use Mermaid's documentation to figure out how to use each arrow in Mermaid syntax.

Submission

The submission should be one PDF file that contains (in order) your sketch, list of actors, list of subcomponents of the system, your requirements, your specifications, and your UML Class diagram. This submission is on Canvas. You can either screenshot your UML class diagram for your PDF or paste the mermaid code itself in your PDF.