1 Pitch 1: Marine Biotope Simulation

1.1 Overview

This is an educational game that simulates a home reef aquarium. It runs in accelerated real time and seeks to educate hobbyists on how to properly maintain a reef aquarium. The game has the following features:

- Choose what kind of aquarium this is (mangrove, SPS-dominant, LPS-dominant, mixed reef, fish-only, etc.) and determine its dimensions, substrate, and location.
- Choose from a selection of common marine fish, corals, and anemones suitable for home aquariums, with accompanying data on their natural habitat and recommended aquarium setup: tank size, is it reef-safe, adolescent and adult size, compatible (recommended) tank mates, etc.
- Choose from a selection of common reef tank equipment: light fixture, wave maker, peristaltic pump, calcium reactor, CO2 scrubber, dosing solutions, etc.
- Choose maintenance schedule. If schedule differs greatly from recommended values, suggest fixes.

The game will simulate an aquarium with the user's selection of livestock, equipment, and maintenance schedule. The main engineering goal is to come up with an algorithm that takes into account changes to each of these parameters to figure out their effects on the continued success of the aquarium.

1.2 Research

These need to be expressable via algorithms:

- Growth rate and pattern of select Acropora species (five).
- Flow pattern of anemones depending on water flow speed/direction
- Rate at which nutrients such as NO_3^- and PO_4^{3-} are depleted by bacteria. This should be dependent on exposed surface area, fish size and quantity, coral colony size and quantity, etc.
- Swimming pattern of select fish species (five)

2 Pitch 2: Smart Home App

2.1 Overview

Create an iOS app that is the frontend to a scalable system with public APIs for third parties to add smart home products. The APIs support things like smart aquarium lights, temperature probes, par meters, etc., that are part of an isolated and integrated environment (e.g. a home aquarium, an aviary). The system uses the Thread protocol to create a mesh network between all connected devices and allows the user to create automation schedules and scenes that can be exported to Apple HomeKit. Here's an example of one such schedule:

- The light begins its daylight cycle at 06:30
- The light reaches its maximum intensity for the day at 12:00
- The light starts to wind down at 16:00
- The light begins its moonlight cycle at 20:30

The system offers automation suggestions to new users. Sample suggestions to be included in the app are

- lighting schedules based on daylight/moonlight cycles in some of the biggest coral reefs in the world (GBR, Red Sea...)
- wave maker settings based on what kind of corals are present in the aquarium
- ozone generator schedule based on the recommended oxidation-reduction potential (ORP) of the water

The system's strength is in its integration of all connected devices. In terms of lighting, this means the system can estimate how much par the lights are outputting based on existing automation schedules and automatically compensate for diode failures by ramping up adjacent diodes or fixtures. An ambitious extension of this feature gives users the option to input the aquarium's general stats like glass placement and thickness, fixture distance from water surface, etc., to determine light spill from adjacent fixtures or reflection against glass caused by fixtures close to the edge. This will be immensely helpful in determining the optimal fixture/diode density and placement so the appropriate amount of par can be achieved.

2.2 Final Product

For the capstone project, I will build the following:

- The APIs to support light fixtures, wave makers, and ozone generator.
- Companion iPadOS app that supports scheduling, spectrum graphing (of the lights), automation suggestions. The app will create a mock Thread mesh network.
- Virtual devices that can be added to the app to test aforementioned functionalities.

3 Pitch 3

Create a smart electronic device (maybe a light bulb). The device will be paired with a Thread chip, a Zigbee chip, a Bluetooth chip (capable of Bluetooh Mesh), and a WiFi chip. The goal of the project is to compare how the device functions under different circumstances with each connectivity standard and collect real-world statistics of how these standards fare against each other. The metrics I'm most interested in are as follows:

- Payload Size: Some applications of smart home technology like security cameras prefer WiFi because it's always connected and the maximum bandwidth is only limited by the WiFi chip used. The result should include an overview of the recommended environments for each standard and provide the exact ranges of framerate and video resolution they can handle.
- Energy Efficiency: Small sensors usually forgo WiFi because they are designed with energy efficiency in mind. In this experiment, I will mostly focus on comparing energy consumption of each LE (low-energy) standard and how environmental factors such as signal strength and distance impact consumption.
- Development Time: For newer standards such as Thread that integrate well with the Matter standard, resources in the form of product samples and firmware examples are relatively scarce. In addition, each standard has their own chip and notarization requirements. I want to detail to interested parties the general cost of development in terms of money, time, and difficulty.

These metrics should help developers of smart home gadgets decide on the most appropriate standard with regard to their intended product category.