

Marine Biology Simulation

Eric Cheng, Dung “Zung” Nguyen, Mathias Ricken, Stephen Wong
Dept. of Computer Science
Rice University
Houston, TX 77005
+1 713-348-3835

ericc@rice.edu, dxnguyen@rice.edu, mgricken@rice.edu, swong@rice.edu

ABSTRACT

The Marine Biology Simulation is designed as a final project in an objects-first CS2 course. It provides an entertaining setting that serves as compelling example of the powers of object-oriented design and programming.

Categories and Subject Descriptors

D.1.5 [Programming Techniques]: Object-oriented Programming

General Terms

Design.

Keywords

CS1/CS2, objects-first, design patterns, abstract coupling, loose coupling, pedagogy, assignments, software quality.

1. INTRODUCTION

Our submission describes the Rice Marine Biology Simulation, an assignment intended to be used as final project in our institution’s objects-first CS2 course that will be taught for the first time in the Fall 2004 semester. The Rice Marine Biology Simulation is derived from the Java AP Marine Biology Simulation Case Study¹ used in high school AP computer science courses and places a strong emphasis on design patterns, loose and abstract coupling, and test-driven development.

The Rice Marine Biology Simulation is a GUI application that simulates the behavior of arbitrary fish species in widely varying environments. Both the fish and environment classes can be loaded into the program at runtime.

2. ASSIGNMENT DESCRIPTION

We propose the following Marine Biology Simulation as a framework for an exercise near the end of an objects-first introductory curriculum. The project teaches students how to use design patterns to correctly model abstraction and to achieve loose coupling between cooperating objects, creating a correct, robust, secure, flexible, and extensible system. Using an

incremental, test-driven approach, the assignment first familiarizes students with the existing framework, subsequently asks them to rewrite critical portions of it, and finally lets them extend it to achieve additional flexibility.

3. WHAT MAKES IT SO NIFTY?

This assignment is interesting and useful for several reasons:

- The exercise teaches students to work with a large, non-trivial project.
- The flexible and secure design is applicable to other challenging programming situations, like operating systems.
- The test-driven development via unit testing provides immediate feedback regarding the correctness of their work while maintaining the freedom to implement code in different ways.
- The project is entertaining and offers an attractive visual component.
- The visual nature of the project enables students to experience the design’s flexibility and extensibility in a compelling manner.
- The assignment requires students to integrate a broad spectrum of skills and concepts (see skills list in section 5).
- The exercise focuses on the thought processes involved with the design of OO systems.

4. TARGET AUDIENCE

This project is intended for students near the end of the second semester of an objects-first curriculum where they have already seen polymorphism, design patterns and component framework systems (see list of prerequisites in section 7).

5. IDEAS AND SKILLS INVOLVED

The Rice Marine Biology Simulation is not about fish. Instead, it teaches students some of the bigger concepts in software engineering, such as:

- Using design patterns to capture appropriate abstractions.
- Creating loose and abstract coupling between cooperating objects in the system to achieve correctness, robustness, security, flexibility, and extensibility.
- Using unit tests as specification.
- Using and building components and frameworks.

- Using refactoring to both simplify and enhance the framework.
- Using abstract factories to manufacture products whose concrete type is immaterial to the client.
- Using anonymous inner classes to define concrete instantiations, especially in the context of commands.
- Using the closure properties of inner classes to directly access both instance and final local variables.
- Numerous compelling demonstrations of polymorphic behavior, especially those that cannot be replicated with conditional statements.

6. LENGTH OF ASSIGNMENT

The project is divided into two milestones. The supporting materials for Milestone 2 include sample solutions for the assignments from Milestone 1.

Students typically have about a week and a half to complete each milestone. In terms of work hours, our estimates for the individual parts are below:

- Milestone 1:
 - Part 1, KnightFish: 2 hours
 - Part 2, WrappingEnv: 3 hours
- Milestone 2:
 - Part 3, Internals: 3 hours
 - Part 4, Behavior Strategy: 2 hour

7. PREREQUISITE MATERIAL

This project is intended as a final project at the end of an objects-first two-semester sequence. It assumes the following material has been covered in class:

- Javadoc and UML diagrams
- Abstract structure and behavior



- Polymorphism
- Using both inheritance and composition as means of extending functionality
- Delegation model programming
- Design patterns, particularly MVC, visitor, command, factory method, and decorator
- Both writing and utilizing unit tests as an integral part of the development process
- Syntax for anonymous inner classes

It is important that this assignment be given in the context of a comprehensive instruction on object oriented programming that stresses abstract decomposition.

8. DIFFICULTIES TO WATCH FOR

Instructors using this assignment should be aware of the following difficulties:

- Understanding the concept of a local environment and how to model it
- Understanding the callback-style communication between a fish and its environment
- Understanding the communication aspects of closures in anonymous inner classes.

9. INSTRUCTION MATERIALS

The assignment, a demo, and the solutions to the different milestones have been submitted to the Educator's Symposium.

10. REFERENCES

1. **The College Board, *Java Marine Biology Simulation Case Study***, <http://apcentral.collegeboard.com>

