ICS 52: Introduction to Software Engineering

Fall 2004

Instructor: Dr. Richard N. Taylor

TA: Justin R. Erenkrantz

Assignment 2: Architecture and Component Design

Issued: Monday, October 18th, 2004

Due: Monday, November 8th, 2004 (beginning of discussion)

Architecture and Component Design Assignment

The Requirements Document that you just submitted has been tentatively accepted by Phoenix's management. While they review your requirements specification, they now ask you to begin the process of documenting the architecture and module design of the prototype. Your focus must now shift towards **how** the prototype works rather than what it should do. Luckily, the developer who wrote the initial *Klax* prototype was a big believer in architectural styles. Therefore, you should find that the prototype largely follows the *C2 architectural style*. By understanding and leveraging the constraints imposed by the C2 style, you should be able to produce the necessary documents. Additionally, Phoenix management has provided a list of additional requirements that their user focus groups have suggested as mandatory improvements in order for the game to be a success in the marketplace. (See the Addendum for the additional requirements.)

Your next deadline is **Monday, November 8th** and Phoenix is looking for a detailed architecture and module design document that fully describes **how** the game should satisfy the requirements. In order to help determine what the game should do, Phoenix has provided you with the source code from the previous developer. You may run the software and "reverse-engineer" the architecture and module design. Some employees remain at Phoenix who know how to run the software, but know little else about the design. Your project manager (aka TA) will hold twice-weekly meetings to guide you through the use of the initial prototype and source code and the creation of the design document.

Structure of the Architecture and Component Design Document

The Architecture and Component Design document should have the following structure:

1. Table of Contents

- Listing all relevant sections and their page numbers

2. Introduction

- A short introduction of the system
- What is this document about?
- Who was it created for?
- What steps were involved in discovering the information contained in this document?

3. Architecture

- Explain the main architectural style used by the system
- Describe the relevant style constraints as they apply to the system
- Provide a rationale, including advantages and disadvantages, of this particular style
- Present a high-level system architecture diagram depicting all components, connectors, and their topology

4. Event Model

- Describe each event that can be produced by the system:
  - What are the properties (attributes) of the event?

  - Which requirements does the event satisfy?
  - Which components generate these events?
  - Which components receive these events?

5. Component Design

- On a separate page for **each** component, describe each component present in the system that addresses the following questions:
  - What is the purpose of the component?
  - What is the generic purpose of the component?
  - Which requirements does it satisfy? If the component satisfies a new requirement, please denote it accordingly.
  - Appropriate items to include: What secret does this module hide? Does the component represent an abstract data type?
  - Which notifications does the component generate?
  - Which requests does the component generate?
  - Which notifications does the component receive?
  - Which requests does the component receive?
  - Are there any additional constraints on the module?

6. Architecture Diagrams

- Include an architecture diagram representing the current state of the prototype
- Include an architecture diagram representing the proposed final state of the system that addresses all additional requirements.

~~7. Integration Test Plan~~

- Organization of tests
- Format of test cases

8. Definitions of Terminology

- Precise definition of terms used throughout the project

9. Reference documents

- Pointers to existing literature and tools
- Pointers to other similar software

Submission Criteria

You **must** submit a hard copy at the beginning of discussion that follows these guidelines, and you **must also** submit an electronic copy to the course EEE DropBox.

Printed Criteria

The printed copy of your architecture and component design that you turn in for credit **must** include:

- Page numbers at the bottom of each page

- Double-spacing for all paragraphs
- Use proportional font similar to either Times (New Roman) or Arial.
- Major section headings in 14 point, subsection headings in 12 point, and body text in 12 point.
- One inch margins around the page
- (**Optional**) Duplex (two-sided) printing is acceptable
- Stapled once in the upper left hand corner, no binders, no plastic covers.
- A title page using a 18 point font with the following text centered vertically and horizontally:

Klax Architecture and Component Design
by:
Your_First_name Your_Last_name
{UCINetID: i.e. panteater@uci.edu}
ICS 52
Instructor: Dr. Richard N. Taylor
Fall 2004

Electronic Criteria

You must submit your electronic version of your requirements document via the EEE DropBox entitled: ***ICS 52 HW #2***. The documents **must** be compressed via ZIP before submission. Word or PDF documents accepted. Other formats accepted by **prior arrangement only**.

Grading

The architecture and module interface design should be clear, easy to understand, easy to implement, and easy to change. A design "that just works" but is not consistent with the design principles presented in class will not get full credit. Apply the software engineering principles introduced in the lectures and found in the textbook.

The grading of this assignment will be broken down as follows:

20%. Consistency to the prototype implementation

20%. Completeness and accuracy of the event descriptions

40%. Completeness and accuracy of the module descriptions

10%. Completeness and accuracy of the architecture diagrams

10%. Incorporation of the new requirements into the architecture and module in a manner consistent with the design principles discussed in class and already present in the prototype.

As a whole, this assignment counts **18%** towards your final grade for the course.

Note:

- Do not work in teams to complete this assignment
- No late assignments will be accepted