Project Phases: applet-based photo album (and more)

March 28, 2007

New - 28 March 2007 — homework 8, due Friday 6 April

Introduction

Summary

Now that each student has a role in a team, the next several homeworks are progress milestones in the development of your project. A reasonable goal for each team is to have a primitive working project by 20 April (the fallback version), including the composer and the player. Instructors can monitor each team's progress by setting subgoals, called *phases* of the project development.

Phase Concept

Standard methodology for product development is a *cyclic* structure. Many industrial products (not just software) have *product life cycles*\(^1\). In software, we have the *waterfall model*\(^2\) of development. Briefly, a development cycle consists of a design and planning stage, followed by an implementation stage, followed by testing. Then you discover how to improve the product, and the cycle repeats. In other words, software is produced iteratively.

We'll try something like the development cycles for the team projects; the work is divided into *phases*, where each phase has the goal of producing something that works. Before the 20th of April, we might have time for three phases.

Phase 1

The goal of Phase 1 is to produce composer, player, and backend for an album of empty pages.

\(^1\)http://en.wikipedia.org/wiki/Product_Life_Cycle_Management  
\(^2\)http://en.wikipedia.org/wiki/Waterfall_model
• There is only one album.

• Initially, the album is empty.

• The composer program can add pages to the album.

• The album can contain any number of pages.

• Each page has a title (a String value).

• The composer program can delete a page from the album.

• The composer program has a way to save the current state of the album.

• When started, the composer program recovers the most recent saved state of the album.

• The player program can display pages of the album.

• The player can step forward or backward through the album.

• The player shows the title of each page when displayed.

The list above has the external requirements of Phase 1. There are internal needs as well. The list that follows gives some ideas about what the internal goals should be for Phase 1 (internal because they are visible in source code and design documents, but not in the behavior of the composer or play applications).

• There are backend classes, interfaces, and methods.

• The backend implementation can translate objects into Strings which are saved to a file (use object serialization for this).

• The composer and player applications use interfaces, defined for the backend classes, to add, delete, and modify the backend objects.

Phase 1: How to get Started?

It will be helpful to have at least one team meeting to discuss the requirements and design. Who will specify (and write code for) the interfaces? What needs to get done first?

Initially, it seems that all development depends on the backend — the composer and player need the backend interfaces to create, destroy, and manipulate album objects. But actually, in Phase 1, both composer and player can start without any objects: these roles can research and experiment with applet technology, learn how to start and stop an applet, learn how to present forms or input boxes to get text from a window, learn how to add buttons, react to mouse events, try pull-down menus, and so on — all of this can be done while the backend is
being implemented. As soon as the backend has the necessary functions working, Role 1 can use this to put the composer together. Further, if Role 3 did a good job testing, there may be already a test album available (built during testing of the backend code) that the player can experiment with.

Homework 8

Due 6 April 2007. This homework has a written component and an electronic component. The written part is done by one team member: the team scribe for the assignment. The team scribe should take summary notes at team meetings, to note what has been decided, who will do what, list outstanding issues (problems, unknowns to be researched, etc.) and near-term plans. Also, the team scribe should report on what has been entered into the team’s CVS repository, the current state of the program (works, works partially, and so on). In Phase 1, we’re not saying precisely what is the format or detailed requirements of the scribe’s report - we may refine that later.

Each team has a CVS repository to share code. The repository path is

/group/class/c022/Team1

for Team 1; the path is

/group/class/c022/Team2

for Team 2, and so on.

The second part of this homework is: each team member checks in the code for Phase 1, in accordance with the role.