

COSC346 Assignment 2: PDF presentation tool

Submission information

Due date: Friday, 6th October 2017, at 5 PM.

Weight: This assignment is worth 20% of the mark for the paper. This mark will be formed from a maximum of 100 points that are awarded as explained below.

Marking: You are permitted to do this assignment individually or in pairs, but the marking will be different in each case.

Submissions from individuals: The *essential functionality* will make up 80 points of your assignment mark. Up to 20 points will be awarded from the items of *additional functionality*. You can accumulate points up to a maximum of 100.

Submissions from a pair: Unless specifically requested by both members of a pair, the same mark will be awarded to both pair members. The *essential functionality* will make up 60 points of your assignment mark. Up to 40 points will be awarded from the items of *additional functionality*. You can accumulate points up to a maximum of 100.

What to submit:

- A complete Xcode project including the source code and all the resources that you use.
- A report in PDF format. (See instructions below.)

How to submit: In the same manner as for assignment 1, collect everything that you need to submit into a directory, and then within MacOS on a Lab machine, run:

```
/home/cshome/coursework/bin/submit346 directory_name
```

A submission from one member of a pair is sufficient if you choose to work in a pair. Late submissions will be handled in the same manner as for assignment one.

Be careful to ensure that your submission only relies on files that you have actually submitted. For example, in the past students have submitted projects that referred to images using file paths that were not included in the collection of files that they submitted.

Goals

The goal of this assignment is for you to design and implement the GUI application described below. The specification is not entirely strict and precise: you will need to develop your own design to meet the stated requirements.

Successfully implementing the application is important, but you should also pay careful attention to the object oriented principles used within your code, and the usability of your GUI. These factors should be discussed in your report.

Problem Description

In this assignment you will design and implement a tool, using Cocoa, for doing lecture presentations from a set of PDF files. You can assume that there is one PDF file for each lecture, but you must allow the teacher (*i.e.*, the user of your software) to open multiple lectures' PDF files together.

In addition to displaying the lecture PDFs, the purpose of this application is to allow the teacher using the software to record short notes connected to particular pages of the PDF content, as well as notes related to each PDF file overall. While the means should be provided to record notes in plain text about particular pages of any of the PDF documents, and also about the PDF documents themselves, persistent storage of this data is not essential functionality.

Also, the teacher needs to be able to record and jump to bookmarks so that they can move between different topics and different lecture PDF files.

When presenting the PDF files, consider in your design that it should be possible to view a window showing just the PDF content. For example, if using a laptop that is plugged into a video projector, this would facilitate moving the PDF window onto the projector's display, while keeping notes and control windows on the laptop's display. Those watching the PDF content should not see unnecessary controls. However, you do not need to test your code on a multi-monitor system: it would be sufficient to allow the user to drag the windows to the appropriate displays. It may be useful to be able to open more than one window containing the lecture content, *e.g.*, so that one copy can be placed on a laptop's display, and another copy shown through a connected data projector.

You are welcome to design an application that has separate organising and presenting modes of operation, or to have one mode that allows for both presenting PDF lecture material and organising it. However, when presenting the PDF content it must be possible to at least view the notes, and use bookmarks, even if at that time notes cannot be edited and bookmarks cannot be changed.

Essential functionality

Your lecture presentation tool must use a design that allows you to provide all of the essential functionality items shown in the list below.

- Users can navigate through a given lecture PDF file, including moving to the next page, moving to the previous page, and jumping to a specific page number.
- Users can navigate between different lecture PDF files within a set, providing both “local” navigation, such as “next lecture” and “previous lecture”, but also the ability to focus on a specific lecture.
- An indication is provided of the current lecture being viewed, and the current page of that document that is being viewed.
- Your application provides functions for zooming in, zooming out and zooming to fit the PDF document contents in the window.
- The application's controls should resize in a sensible manner when its containing window is resized.

- Users can record brief textual notes that are related to a particular page of a particular lecture PDF file.
- Users can also record brief notes that are related to a particular lecture's PDF file.
- Users can bookmark particular pages within particular lecture PDF files and then later use these bookmarks to jump back to the appropriate page within the given PDF file.
- A “useful” menu structure is implemented, that complements your other user interface controls.
- Your “About” panel must be customised to include some relevant information about your project, *e.g.*, giving credit to the creators of any resources that you acquire from elsewhere, such as icons.

There are many ways that you can achieve the navigation and display functions discussed above. While you are encouraged to be creative with your user interface, do remember that you will be marked on the usability of the system, so your design choices must be justified.

The core document display requirements of your PDF lecture presentation tool should be met by Apple's PDF Kit. Displaying PDF content will mostly involve calling methods of instances of the `PDFViewer` class.

The PDF Kit documentation is available via https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/PDFKitGuide/PDFKit_Prog_Intro/PDFKit_Prog_Intro.html although note that it gives Objective-C examples—you will have to determine how to make equivalent method calls from Swift. In contrast, the API documentation provided at <https://developer.apple.com/documentation/quartz/pdfkit> gives details for both Swift and Objective-C.

Check the COSC346 website for accompanying documentation: directions will be provided to show you how to integrate PDF Kit support into your project, and also how to create a project that does not use storyboards. (You are welcome to use storyboards or not, although the lecture material typically does not use storyboards.)

Additional functionality

- Implement persistent storage of the notes made on PDF files, and on the pages within them. **[10 points]**
 - Your application should be able to write all of the required application state to disk and to be able to restore the notes from disk.
 - Loading notes back from disk should (try to) open the associated PDF files.
- Implement rich formatting of the notes made on the PDF files. **[10 points]**
 - The formatting must not only apply to a whole note at a time: parts of the text *within* each formatted note must support having formatting applied to it.

- Provide support for use of PDF annotations, for example to support additions made during a lecture by the teacher. **[20 points]**
 - Subclasses of the `PDFAnnotations` class are likely to be useful.
 - Implementing two types of annotation is sufficient.
 - You should provide undo and redo functions for annotations.
- Add large displays of the time of day and the elapsed time within a lecture. **[10 points]**
 - Appropriate controls will need to be provided to start, pause and reset the elapsed time.
- Add an unattended presentation mode that progresses the slides automatically, pausing on each one for a specified time. **[10 points]**
 - Different slides will need to be able to support different pause times.
- Add document search capabilities. **[10 points]**
 - PDF Kit can provide the search facilities that you would need to use.

Use of external resources

If you wish, you can design your own icon for your application, but you are also welcome to use any freely-licensed external graphical content, such as the set that is available at http://www.gentleface.com/free_icon_set.html#geticons. You must include relevant credits and licensing information in your “About” panel. Also, be sure to test that these resources are included and correctly linked into the Xcode project that you submit.

Report

You must provide a report in PDF format that justifies your key GUI and object-oriented design decisions. Your report must also list the features that you chose to implement, and how users should use them—particularly where you have implemented keyboard controls or short-cuts. This is important for the marking of the assignment: you want to ensure that none of your hard work is missed. You do not need to go into detail explaining the internals of your implementation, as the comments in your code are likely to provide a useful guide on that front.

The report usually should not need to be longer than one page. You will lose marks if the report has obvious typos, spelling or grammatical errors. Submissions from pairs must also explain the role taken by each member of the pair in completing the assignment.