CS193E: Assignment 4
PersonalTimeline I

Due Date

This assignment is due by **11:59 PM, February 6**.

Assignment

Over the next four weeks you will build a full-featured application including multiple document support, copy/paste, drag/drop, mutlti-level undo, and a custom view. As always, the journey of thousand miles begins with a single step. This week, we will take that first step.

The Personal Timeline application will allow a user to enter a series of events, and display them in a timeline view. This week you will lay the foundation by creating a document-based application. You will create the model for the application—a timeline which has an array of timeline events—and integrate the model with the document. You will display the events in a table view in the document window, with buttons to add and delete events from the table. Finally, you will implement multi-level undo and redo.

There are a number of parts to this assignment. Here's a rough outline of what you'll need to do to complete the assignment successfully:

1. Create a new document-based Cocoa application called "PersonalTimeline" in Xcode.
2. Set the document type to `.timeline`, following the steps we discussed in class.
3. Create and implement classes for your model. In this case, you'll create a `Timeline` and a `TimelineEvent` class. The `TimelineEvent` class holds the information about a single event in a timeline. It should implement methods for setting and getting a title, a date, and a numeric rating from 0 to 5. In addition, it should know how to encode and decode itself.

   The `Timeline` class holds an array of timeline events. It should implement methods for inserting and deleting events at a given index, getting the count of events it holds, and getting a event at a particular index. It should also know how to encode and decode itself.
4. Create and implement a class for your controller object. (In this case, this is a subclass of NSDocument called MyDocument; it's created for you automatically when you create the new project.) The MyDocument class should implement the methods needed to save and load a timeline, and create a new timeline if there is no timeline to load.
5. Add a table view to the user interface. Set the data source of the table view and implement the table view data source methods to allow display and editing of values in the timeline's events. The table view should have 3 columns, one for each piece of information held in the event (title, date and rating).
6. Add buttons to add and delete events. Add actions to your document subclass to handle these operations.
7. Implement undo/redo. You should be able to undo adding and removing timeline events as well as changing values of the properties of a timeline event.

Testing

Make sure the following work in your application (these are the things we will test to determine your grade). Many of these should come "for free" with the NSDocument architecture we are

leveraging. However, *make sure to test them all* as some of them require a little bit of work on your part.

1. Your project should build without errors or warnings.
2. Your project should run without crashing.
3. Make sure multiple documents can be created simultaneously with `File→New`. The document windows should be staggered, and have incrementing names ("Untitled", "Untitled 2", "Untitled 3", and so on)."
4. Events should be displayed in a table view with three columns
5. Clicking "Add" button should add a new event
6. Clicking "Delete" should delete the selected event
7. You should be able to edit the value of the title, date, and rating.
8. The date column should use a text field and show a formatted date.
9. The rating column should use an NSLevelIndicator cell to show ratings.
10. Make sure edits made to one document do not affect any other open documents.
11. `File→Save/Save As` should save the document to the specified file, and the extension should default to `.timeline`.
12. File→Open should restore a document completely, including the event order (i.e., the events should be listed in the same order as when they were saved). The Open panel should only display documents with the `.timeline` extension.
13. File→Open Recent should track recently opened/saved documents and should cause those documents to be loaded when invoked.
14. You should be able to undo/redo additions and deletions of events from the timeline.
15. You should be able to undo/redo changing properties of the events.
16. When you've undone all the undo operations, your document's dirty indicator (the dot in the middle of the close button on the window) should not be shown.
17. The Undo/Redo menu items should have descriptive names to give details about what they are undoing and redoing.

Hints

Try getting everything except undo working and tested first. Then tackle undo.

It is recommended that you keep the MyDocument names for the document nib and the NSDocument subclass; renaming them will prove to be a surprisingly challenging distraction.

Make sure the accessor methods you write follow the Key-Value Coding naming conventions. This is both for simple properties, but also for the array of timeline events.

NSTableView has a large set of methods. One that will be particularly helpful for this assignment are -reloadData which tells the table view to refresh itself after you've made a change. Another useful method is -selectedRow which will return the index of the selected row or -1 if no row is selected.

To help in implementing undo/redo, your document class should serve as a funnel point for changes in the model.

Troubleshooting

Some common problems when using a table view are forgetting to hook up the data source (either in Interface Builder, or programatically) and misspelling the name of a data source method. If you are using Key-Value Coding, be certain that the table view identifier string set in Interface Builder is the correct key for the property that column is displaying.

Another common mistake when using table views is to have a misspelling in the table column identifier. One way to minimize this risk is to create string constants (either #define or a static NSString variable) for the column identifiers and use those constants in your code.

When working with an NSArray, you should be very careful with the indexes. NSArray will throw exceptions if you try to use an index that is out of bounds. Before using an index, you need to verify the validity of the index.

Extra Credit

Keep in mind that extra credit only applies if the required behavior outlined above is working. Don't spend time on extra credit until you've got the basics down.

- Æsthetics matter: a particularly attractive, innovative, or clever design of the interface is always appreciated. (If you want consideration for extra credit in this case, you must let us know when you submit your project.)
- Add a Timeline menu and menu items to add and delete an event. Note the delete menu item should only be enabled if the table view has a valid selected row. Also remember that these menu items should target whichever document the user is currently working on.
- The delete button currently stays active even when nothing is selected in the table view. You can register for a notification to find out when the table view's selection has changed. Use the selection changed notification to update the enabled state of the delete button to only be enabled when deleting is a valid action.