CS 4063/5063
Homework: Design E
Due Thursday 2022.04.14 at 11:00pm.

All homework assignments are individual efforts, and must be completed entirely on your own.

In this homework you will: (1) gain experience composing and styling basic wireframe elements; (2) create a wireframe of a Cover Flow to add to the Collection Pane, based on a UI specification; (3) storyboard an interaction sequence for typical navigation in the UI; (4) do a Fitts' Law comparison of navigation in the Cover Flow and Data Grid; and (5) document how you styled color & text to convey movie attributes in the Cover Flow.

Reminder: References for Using Wireframes

user guide: https://balsamiq.com/wireframes/desktop/docs/
tutorials: https://balsamiq.com/tutorials/
intro to wireframing: https://balsamiq.com/learn/courses/wireframing/

The Design Scenario

Imagine you are a designer on a large team. The team is developing a new application in JavaFX for browsing a personal movie collection on a desktop or laptop. The team is currently exploring ways to integrate a more usable, efficient, and appealing mechanism to browse the collection of movies into the Collection Pane. You have been tasked with designing the appearance and behavior of a Cover Flow and the individual movies in it. The team has asked you to provide a complete description of the Cover Flow layout, the graphics shown for each movie, and how the layout animates in response to interaction.

The Cover Flow Design Specification

“The Cover Flow will display movies in the collection as a navigable sequence of items, each with graphical styling and decoration to convey key attributes of its movie. Assume an available area of 640x180 pixels. The Cover Flow must display at least 5 and at most 11 items at a time. The size of each item must be between 120 and 160 pixels in each of width and height. Each item must include all of the details specified in the table below for the corresponding movie, accentuated by visual importance. The currently selected item must be horizontally centered with the other items laid out to the left and right. The items may overlap (judiciously!) except for the centered item, which must be in front of all others. The currently selected item must be differently styled to indicate that it is the selected one. Styling of layout and items is otherwise left to the designer's discretion.

The Cover Flow will support interactive navigation via buttons on the left and right sides. Each side will layout three buttons: one button to move left/right one item, one button to move left/right five items, and one button to jump to the first/last item. The buttons may appear above any visible items. Each item will also support mouse click interactions; the clicked item will become the selected one. Whenever a movie is selected—by whatever means throughout the application UI—the Cover Flow will animate by sliding all items left or right until the newly selected item comes to rest in the horizontal center position."

Movie Attribute	Display As	Importance	Suggestions
poster image	image	very high	browse `javafx.scene.effect` for styling ideas
title	text	high	okay to use wrapping and/or ellipses for long titles
genres*	colors*	medium	explore qualitative color schemes at colorbrewer.org
rating	shape	medium	browse `javafx.scene.shape` for geometry ideas
average review score	length	low	length of a line, height or width of a rectangle, etc.
one boolean attribute	*your choice*	low	*what are good ways to show true v. false graphically?*

Wireframe a User Interface for Cover Flow Layout

Think about what each movie attribute means and how it relates to the others. Drawing from your experience with how apps and documents can be displayed as a combination of icons, labels, badges, borders, etc., decide which specific visual characteristics you will use to display each movie attribute, following the specification and the table above. *(*Associate each possible genre with a color. Incorporate each movie's genre colors into its item design somehow. Remember, a movie can have zero, one, some, or all genres.)*

In Wireframes, create a new project called DesignE. Name the wireframe Cover Flow 1. Import the movie poster image files from any of the PrototypeX assignments as assets, then create a 640x180 ***Rectangle*** element to contain your design. Fill in the rectangle to display your layout of navigation buttons and a representative set of movie items. Style the various elements as best you can using Wireframe features. The ***Label*** and ***Shape*** elements are likely to prove especially useful for this! Use the built-in collection (Font Awesome Icon Set) for any icons you need in the design. You are also likely to want to apply various styling choices that Wireframes doesn't support. For those, add nearby Markup elements to briefly describe the styling you have in mind for the implementation. Carefully layout and style all elements in your design, keeping in mind that you will use JavaFX to implement the design with interaction as precisely as you can in PrototypeE.

Storyboard a Common Navigation Task

Users will interact with the Cover Flow by clicking buttons and items. Suppose that the user performs the following interactions to browse movie information in the Cover Flow: (CF1) start with the first/leftmost movie selected; (CF2) click the button to navigate to the last/rightmost movie; (CF3) click the button to navigate backward/left by five movies; (CF4) click an item to navigate forward/right by two movies; (CF5) click the button to navigate back to the first/leftmost movie. (If you include all 22 movies from the example data set in alphabetical order, the movies visited should be *2 Fast 2 Furious*, *Zoolander*, *The Chronicles of Riddick*, *The Pursuit of Happyness*, then back to *2 Fast 2 Furious*).

Create a duplicate of your Cover Flow 1 wireframe for each of steps 2–5. Rename each to Cover Flow # using its step number, then alter it to show what the UI would look like just **after** the user performs the corresponding step. Add Markup elements to help your team visualize each interaction and the corresponding animated transition. In each wireframe, link the element acted upon to the next wireframe. Link Cover Flow 5 back to Cover Flow 1 to allow restarting. (For elements that don't support the "Links" option, you can add a *Link* element on/near the element.) Use Full Screen Presentation mode to replay the task. *Do not include a Task Flow diagram, home links, or back links this time.*

Analyze and Document the Design

Use Fitts' Law to calculate the expected time for a user to complete the task using the Cover Flow. Calculate the same using the unfiltered version of the Data Grid/Table from DesignB. Analyze and compare how their respective navigation interactions affect task efficiency. Assess tradeoffs in your Cover Flow design between efficiency and at least 4 of the other 9 principles of usability from the slides. Add a Comment to the Cover Flow 1 wireframe with your calculation details, efficiency analysis, and tradeoffs assessment.

Finally, add another Comment to the Cover Flow 1 wireframe to explain the rationale behind your specific choices of colors and text styling of movie attributes in Cover Flow items. Support your explanations with reference to the technical considerations and design guidelines offered in the *Color & Vision* and *Text & Readings* slides from class.

Turning It In

Use the Project/Save Project As... menu item to save a copy of your project. The exact name of your file should be simply `DesignE.bmpr`. Make sure it contains all and only the wireframes, assets, links, markup, and comments that you wish to submit. Submit the file to the Homework - Design E assignment in Canvas.

To score the assignment, we'll be looking at the completeness and quality of your Cover Flow design; the appropriateness of your Cover Flow item display choices relative to the specification; how clearly you have conveyed interaction via a sequence of wireframes with Markup; and the cogency, thoroughness, and depth of your calculations, analyses, assessments, and explanations in Comments. The maximum score is 20 out of 20.