CSC 2002S: Mobile Design & Development

Melissa Densmore – September/October 2017

Introduction

You have two assignments for this class. For A5 (Phase 1, due 5 October) you will demonstrate that you can create a basic Android application. For A6 (Phase 2, due 18 October) you will use what you are learning about design to prototype an improved application. Both assignments are to be done individually.

The deadlines for the assignments are very tight and you will **not** be able to get extensions for them. To help you plan and prepare, both assignments are being released and described in this document.

A3 (Phase 1) An Alphabet Book

Due Thursday, 5 October 2017 (by 10h00)

Marks – 30 points

You will implement an alphabet book in Android. Included as an attachment in this assignment is a set of 26 images, one for each letter of the alphabet[1]. There are two main views for the app, an overview page, which has a button for each letter of the alphabet, and a letter page, which will show the image corresponding to the selected letter.

On the letter pages, there should be five buttons:

- First page (‘A’)
- Previous page
- Next page
- Last page (‘Z’)
- Overview

When you open the application, it should always open to the last letter (or page) viewed, unless it was last opened on the overview page, in which case it should open on the overview page. The images should be stored on the phone filesystem rather than compiled into the application under resources. If you choose to use icons for the buttons or the overview page, you may place those in the resources.

[1] Images sourced from Emoji One library, except for Y – ‘yield’ and Z – ‘zipper’, which have been sourced from Wikipedia Commons.

Please submit your source code, and a compiled APK that will run on a Google Nexus 5X emulator running Nougat (API Level 24). All files should be put into a single archive, which should be named according to your student number and the assignment number, e.g. ABCXYZ001_CSC2002S_A3. Upload the file and then ensure that you have uploaded the correct file.

You may develop your app in whatever environment you prefer (Android Studio is recommended). However, you should test that it runs with the emulator that comes with Android Studio. **You must also sign up on Vula to do a five-minute demo with one of the tutors during one of the designated slots before Thursday 12 October.** *Failure to show up for your demo or to successfully demonstrate your app by this date will result in zero marks for the assignment.* If you will not be doing the demo on an android device or on an emulator on a laptop, please be sure to arrive for your demo early so that you can set it up and ensure that it is running on an emulator on one of the available machines.

I highly recommend that you start working on this right away. The biggest barrier in getting this assignment done is often getting your development environment configured and running, rather than the code itself. Please reference the Android Developer tutorials and other resources on https://developer.android.com/training/index.html.

Unknown
Field Code Changed

Grading

Code: 30 marks

This phase will be marked on how well you follow Android conventions and make use of the facilities provided in the Android SDK. You will not gain extra marks for advanced functionality. This phase will be assessed on the submitted apk and code alone.

Category	Description	Points
Demo 10 points	**Does the APK run as prescribed?**	5
	Overview: Are there buttons for each letter, which display the corresponding letter?	1
	Back: Does the app handle the back button appropriately?	1
	Previous Page: Is the button implemented correctly?	1
	Next Page: Is the button implemented correctly?	1
	First/Last Page: Do the buttons take you to 'A' and 'Z'?	1
Code 5 points *based on code review*	**Overview Page:** Is there code/xml for the overview?	1
	Overview Button: Is there code/xml for a button?	
	Previous Page: Is there code/xml for the button?	1
	Next Page: Is there code/xml for the button?	1
	First/Last Page: Is there code/xml for the buttons?	1
Following Conventions 15 points	**Layout:** Is the layout specified in an xml file located in the res/layout directory?	2
	Resources: Icons contained in resources. Strings should be	2

specified in res/values/strings.xml and not hardcoded in the layout or the java files	
Activity Bar: Is there an activity bar, preferably with at least one of the four buttons?	1
MVP Model: Are there classes that clearly correspond to the model and the presenter? Model = Representation of and access to the images View = How the model is shown to the user, the interface Presenter = Manages user input and model updates	3
Activity: appropriate methods should be implemented to retain state (i.e. current image) when app is destroyed, paused, or stopped	2
Images: Images should be accessed by iterating through the either the Pictures or the DCIM folder.	1
Multi-threading: Images should be loaded in the background and not block the main thread	2
Memory Management: Bitmaps should be sampled so the app does not run out of memory rendering hi-res images unnecessarily	2

A3 (Phase 2) Make-Your-Own Alphabet Book

Due Wednesday 18 October 2017 at 10h00

Marks – 70 points

You need to design and prototype a more advanced alphabet book. Users should be able to customize the alphabet book, taking their own photos or using their own images for the different letters. You should start by listing four or more new features or design changes for the system, explaining why you think those features are useful on a mobile device. Do not just give personal opinion - think about the social issues discussed in lectures, and design patterns you may have seen in similar apps.

Next, you need to design how the interface for this functionality will appear. To that end you need to create a prototype and annotate it with a rationale for your design; be prepared to explain why your design looks the way it does with reference to the design ideas in the lecture. Simply stating that you think your design looks nice does not constitute an acceptable rationale. The more (relevant) information you can give about why the interface looks the way it does, the more marks you will receive.

You do not need to implement this as an android application – you may use any prototyping tool, such as Invision, POP, Fluid UI, or PowerPoint. Hand-drawn images may also be acceptable, but you should justify and discuss the fidelity of the prototype.

For this assignment, you will submit a written report of up to 3500 words, containing:

- Overview of the app, describing your target audience and any assumptions you have made. You will want to show how the four features you choose fit together in the overall app.
- A list of features with an explanation for their inclusion
- An initial design for each feature. This will consist of a number of screenshots, with clear markings of how one transitions from one screen to the next. It may be easiest to have a diagram with numbered elements and refer to those numbers in the text. Be sure to show how errors or edge cases would be handled in your design. *Diagrams may be hand-drawn if desired but must be clear to the reader.*
- For each feature, justify the design; explain why it looks and works the way it does. Do not just state the name of relevant design principles, but make an argument for why others will appreciate your design.

GRADING

FEATURE RATIONALE AND DESIGN: 60 MARKS

Each feature is worth 15 marks. Up to four features will be marked.

- **Feature Rationale (5 marks):** Fully justify the four features that you decided to include. Include details of any research or experience that lead you to choosing them with referencing.
- **Design (5 marks):** Document your screenshots, showing how to trigger transitions, and clearly showing the interface for your feature.
- **Design Rationale (5 marks):** Explain your design fully, linking it to the concepts and ideas that you were taught in class. Include in this the ways in which it conforms to the different standards and models, if appropriate. Every design decision should be motivated.

	Not Done (0)	Fair (1-2)	Good (3-4)	Excellent (5)
Feature Rationale	No feature specified.	Feature described adequately but not justified.	Feature justified.	Clearly articulated and cited rationale for feature.
Design Prototype	No design specified, or design unclear.	Prototype provided (of any fidelity) but transitions and where it fits in the existing app are unclear	Clearly depicted prototype including transitions, error handling and synthesis with current app	Innovative prototype, clearly articulated, leverages affordances and mappings in an intuitive way.
Design Rationale	Design not justified.	Justification is given but doesn't reference design principles or other concepts taught in class	Justification includes either an explanation of how it adheres to standards and models or it is linked to HCI and design concepts	Justification correctly references both appropriate standards and HCI design concepts.

OVERALL DESIGN AND LAYOUT: 10 MARKS

Provide an overall design for your finished app, showing how all of your features will interact. Make sure this corresponds to your explanations with the rest of the assignment. You should discuss your target users and state any assumptions. Specifying the device(s) that the app would run on may be helpful too.

Score	Description
Not complete (0)	No layout section or screenshots provided
Poor (1-2)	Screenshots incomplete or screenshots provided but it is not clear how the different features and screens will interact.
Minimal (3-4)	Layout is specified but the features or designs do not mesh well together, or are confusing to the projected user.
Sufficient (5-6)	Interactions between features are clearly specified, with markings for how various features are invoked, and how different user interactions lead to changes in screen layout
Good (7-8)	All of the above, and the screenshots adhere to the appropriate platform design principles (probably android). Clear synthesis of the app overall (i.e. consistent fonts, buttons, etc).
Exemplary (9-10)	Strong discussion and targeting of the designs to a target audience, complete application specification, accounting also for application and user errors.