

## **Mobile Application and Web Design**

### **Final App Design Project**

#### **Description and Assignment:**

We have learned several Android/Java tools for Mobile App Development:

- Android View Objects (Buttons, EditText, SeekBar, TextView, ImageView)
- XML Layouts and Attributes
- Android Sensors (Accelerometer, GPS)
- Handler / Runnable Sequence
- Custom View classes for Animation
- Canvas Operations

Develop an Android Application of your choice. You may choose from the following two options:

1. Choose an Application from the Text Book. You will need to refactor and "fix" any errors the textbook might have to make the application run. You must also add features not shown in text as described in the rubric. (Caution - Apps in the Book do have errors, so please code carefully and be ready to 'debug' and fix errors as they occur. Also, the book apps do not demonstrate the best Java conventions - be sure to consult the rubric to refactor and clarify code as needed.)
2. Create an application "from scratch". It must be a significant improvement over projects and ideas we explored in class.

If you include new features from the Android API or examples of code from other projects - you must cite these examples in comments (author, website or book where you got the ideas from).

Remember - the app must be useful, entertaining, and well developed. Do not wait until the last minute to build/code your project. Use class time wisely and budget out of class time to research, code, test, and debug.

#### **Turning Project In:**

Project should be coded and saved in the shared Android folder on your SkyDrive. Make sure your SkyDrive is shared with the Mr. Michaud and the App runs as you designed. Mr. Michaud will run and test each App in addition checking the code. You will present your App to the class or have an interview/demonstration with Mr .Michaud during the last two days of the term.

**Rubric: (Maximum = 117 Points)**

	<b>Novice (Up to 7 Points)</b>	<b>Emergent (Up to 9 Points)</b>	<b>Meets: (Up to 11 points)</b>	<b>Exceeds: (13 Points)</b>
Use of Control - Model - View Application Design	Only the Activity Class is defined. No modeling of objects or data in program is present. Data passed from sensors directly to view without use of variables or modeling.	Only the Activity Class is defined. Modeling of data in program is present with fields and functions.	Activity and one additional class is defined. Model is evident between the two classes. One class is an extension of View for animation.	Multiple classes present to create a well defined and flexible model for the application. Accessor and Modifier functions exist to pass data through the Application. Objects in App such as characters in game or agents in program are modeled as a separate class.
Android Sensors and Events (Accelerometer Sensor, Touch Sensor, ...)	No Sensors used in application.	Attempted to use Sensor in Application. However, the sensor does not work as designed or does not add functionality to the App.	Sensors are used during application well coded Listener functions or classes. Data from Sensors is modeled and used to control functionality of App.	Sensors are used during application and data from sensors is passed between classes using model with variables and functions. Sensors provide key functionality for the App.
Handlers and Runnables	No use of Handlers and Runnables present in App.	Handler and Runnable sequence present. However, the sequence does not work as designed or does not add functionality to the App.	Handler and Runnable sequence is present. Sequence works to control animation or timed events. Function or functions work to start and stop sequence as needed for function of the App.	Handler and Runnable sequence present to provide key functionality to Application. Multiple use of Handler and runnable present or Handler and Runnable controls the animation and graphics of application.

XML and User Interface Design	Errors present in XML Layout. Minimal or no use of XML layout tools such as tables and table rows. No use of custom view objects in XML.	XML runs as designed. However, the XML does not deviate from basic layout. Less than 3 XML objects present.	XML fills out features for the App, providing a wide selection of buttons, seekbars, imageViews, and custom View Object.	XML fills out features for the App, providing a wide selection of buttons, seekbars, imageViews, and custom View Object. Colors and design of XML interface are visually pleasing and add creativity and functionality to the App.
Relevance and Creativity	App has little relevance beyond fulfilling the requirements of the rubric.	App has relevance to real world use but shows little or no visual creativity beyond the function of the program.	App has relevance and employs some creative touches in visual, sound, or sensor design.	App is fully featured and presents a possible prototype for a consumer or commercial application. Visual and functional elements of the App combine for a compelling user experience.
Java and Android API Conventions	Comments are not used or are not descriptive. Names of Java objects do not match their role in the program. Code does not compile and run. Errors are present in syntax and execution of code. Application crashes during execution. No use of Arrays or ArrayLists in program.	Application runs and does not crash. However, application does not perform as intended. Some comments are used - but they provide little description or guidance as to the intentions of the program. Fields and Functions exist - but are inefficient or do not have descriptive names. Code does not follow Java conventions for names of variables. No use of Arrays or ArrayLists in program. No use of MediaPlayer in	Comments exist and provide description for classes, variables, fields, and functions. Code compiles and App runs as designed. Inner classes exist to retrieve data from User events in the form of Listeners. App features multiple classes with accessor and modifier functions. Proper tabbing and naming conventions help code be easy to read. At least one Array or ArrayList	Comments exist and provide description for classes, variables, fields, and functions. Code compiles and App runs as designed. Inner classes exist to retrieve data from User events in the form of Listeners. App features multiple classes with accessor and modifier functions. Proper tabbing and naming conventions help code be easy to read. Program explores Android API elements not covered in class. Arrays or ArrayLists are used to model data and control animation objects with

		App.	is used in program to model or control data. MediaPlayer and sounds are used in App.	the For Loop. MediaPlayer and sounds are used in App.
Canvas and Animation	No use of Canvas, graphics, or animation exists in App.	Canvas and Graphics are used in App, but without Animation and Handler/Runnable sequence.	Canvas and Graphics are used in App with Animation. However no outside image resources are used with App.	Canvas and Graphics are used with App with animation and outside image resources using a Rect object to scale as needed. Animation and Canvas provide some key functionality for App.
Outside Media	No outside media present in App (Sounds and Images)	At least one sound or image graphic is used in Application.	Multiple sounds and images are used with App without Animation.	Multiple sounds and images are used in App with Animation.
In Class presentation or demonstration / interview with Instructor	No in class presentation or interview is conducted.	In class presentation / interview demonstrates function of App with little explanation or understanding of the purpose of App and function of underlying code.	In class presentation shows understanding of App and underlying code. Delivery of presentation lacks enthusiasm or presence.	In class presentation shows complete understanding of App and underlying code. Delivery is creative and student works to "sell" the App as a viable and interesting product for consumers or business. Creative presentation holds class or teacher's attention.