

Directions:

Using the Online Tutorials, create the following Applications and Run them on the emulator and the Android Device. Create a short JING for each app demonstrating your code and the App running on the emulator.

Project 1: Android Drum/Sound App: (50 Points)

Standard:

At least 8 Buttons with Graphic Images

At least 8 Sounds triggered by the Buttons

Resources:

<http://nebomusic.net/appinventorlessons/drummachine/>

<http://198.211.103.19/~nebomusic/AppInventorMedia/>

Project 2: Android Sign Language App: (50 Points)

Standards for Sign Language Game

Have an Image Object to show the ASL Alphabet

Have a Text Field to enter a Letter or word

Have a Translate Button that shows the ASL sign for the letter entered in the text field

Resources:

<http://nebomusic.net/appinventorlessons/aslapp/>

Project 3: Android Avoider Game: (50 Points)

Standards for Avoider Game

Avoider Sprite whose position and Movement is triggered by User Touch

At least 3 other Sprites for the Avoider to "avoid" during the Game

Game should display Game over when Avoider is hit.

Resources:

<http://nebomusic.net/appinventorlessons/avoidergame/index.html>

Project 4: Your Choice: (52 Points) (See Rubric on Next Page)

Choice 1: Create Your Own Android App:

Choice 2: Take Project 2 or 3 from above and add additional Features such as:

Text To Speech (ASL App)

Sound Effects or other Objects

Scoring or Levels (Avoider Game)

Any other creative modifications to Avoider Game (Different characters)

Using the Accelerometer to Control Game Objects (Avoider Game)

Choice 3: Select App project from Georgia Tech Distance Learning, MIT, or nebomusic.net and Create App:

Resources:

<http://nebomusic.net/appinventorlessons/>

Rubric for App Inventor Project #4

Maximum Points: 52

	Novice (Up to 7 Points)	Emergent (Up to 9 Points)	Meets (Up to 11 Points)	Exceeds (Up to 13 Points)
<i>Outside Media</i>	No outside media present in project. (Sounds and/or 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 with App with Animation.
<i>JING</i>	No Jing present.	Jing is present, but does not describe App Function or Code Blocks.	Jing is present, but only the App Function or Blocks are explained. More explanation is needed.	Jing is present and completely describes App Function and Block Coding. Jing illustrates understanding of program process behind the App.
<i>Objects</i>	Less than 3 Android Objects are used in App. Naming conventions are not followed.	Less than 3 Android Objects are used in App. Objects have proper naming conventions (descriptive names based on purpose or role in Application).	At least 3 Android Objects used in App. Objects have proper naming conventions and properties of objects have been modified. (Colors, size, layout . .)	More than 3 Android Objects are used in App. Objects have proper naming conventions and properties are manipulated by code blocks. (Color, Location, Size . . .) Non-Visible objects such as media players, Sensors, or Clocks are present with Code.
<i>Creativity and Functionality</i>	App project is an attempted exact replica of instructions. App does not function as designed and crashes.	App project is an exact replica of instructions without modification by student. App works as designed.	App project is an extension of the instructions. New objects, images, and function are introduced by student to personalize Application. App runs without crashing.	App project is completely created and implemented by student by synthesizing concepts from the demonstration Apps. Sounds and Images are created or edited by student to

				customize the App. App features programming concepts not introduced in class. App uses variables, arrays/Lists, conditionals, and procedures defined by student.
--	--	--	--	--