App Inventor Drum Machine Instructions (Project #1)  
(Version 2 of App Inventor)

Description:

App Inventor is a web based tool that allows the user to create apps for Android devices. The user interface design is assembled in a web browser and the programming of the app objects are done in a block based environment. App Inventor provides a platform to prototype App ideas and experiment with the elements of Mobile Application design such as buttons, text boxes, events, sound, and animation.

Elements of App Design:

A mobile application has three parts; Control, Model, and View:

![Control-Model-View Diagram](image)

In App Inventor:

- The Control and The View are set with the web based interface (http://beta.appinventor.mit.edu)

- The Model is programmed with the Blocks Editor (A java applet downloaded when needed during session)
User Interface Design in Web Browser:

We design the user layout of the App in a web based application found at: http://ai2.appinventor.mit.edu/. There are four windows in the web based user interface editor:

**Palette:** Contains element such as Buttons, Labels, TextBoxes, Sounds, Sensors

**Viewer:** Contains a representation of the App Screen

**Components:** User Interface elements currently in App.

**Properties:** The specific settings and values associated with the selected element.
Programming The App:

To program the actions of an App, the Blocks Editor. Each block represents an event, value, or function. The example below shows the blocks that will direct the Sound objects to play when a specific button is clicked:

![Model: App Logic](image)

**Event Driven App**: When Button Clicked → Play Sound
Process:


2. You should see a screen that looks like this: Click “New” to create a new App Project.
3. Type “LastnameDrumMachine” in the Project Name Box and click ‘OK’.

4. You now see the User Interface design screen. We need to make a 2 x 4 array to hold 4 button objects. All objects in App Screens must be contained inside a screen arrangement. On the Palette, click ‘Layout’ and drag a ‘Table Arrangement’ to the Screen.
5. We now need to set the properties for the table arrangement. On the right hand side of the screen in the ‘Properties’ window, change the following values:

Columns: 2
Rows: 4
Width: Fill parent (takes up the whole screen)
Height: Automatic
6. We are now going to place 4 Button objects. Click on the ‘User Interface’ tab in the Palette and drag 4 Button objects to the Table Arrangement. Notice how they show in the Components View as an outline:

![Image of MIT App Inventor 2 interface showing a Table Arrangement with four buttons]

7. We need to set the properties for these buttons. We are going to set the following properties:
   - Width: 150 Pixels
   - Height: 150 Pixels

   Click on each button and set the width and the height in the Properties window:

![Image of Button properties window showing Width set to 150 pixels]
The result should look like:
8. We now need to import the Sound objects. The Sound objects will hold the sounds we want to play when we press the buttons. Click on the Media tab in the Palette and Drag 4 Sound objects to the phone screen. Note that you will not see them on the screen, but they will show up in the Components outline.

9. We now need 4 sounds for the sound objects. You may search for sounds (instrument, animal . . .) on the internet or you may use sounds I have set aside for practice on Apps. (Located at: http://198.211.103.19/~nebomusic/media/) To get to the App Inventor Media collection, open a new tab on your browser and go to the Computational Media Page.

On the “Links” section of the page find the link for ‘Media for App Inventor Projects’
10. Once you click on the link you will see a screen listing Backgrounds, Costumes, and Sounds:

![Index of ~/nebomusic/media](image)

11. Click on ‘Sounds’ and select a folder. (I am going to select ‘Instruments’). To listen to the sounds, click the link. The page will change and the sound will play. To get back to the sounds, use the back button.

![Index of ~/nebomusic/media/Sounds/Instruments](image)
12. After you have listed to some sounds. Select one and right click on the link. Select ‘Save Link As’

13. Navigate to your Computational Media / App Inventor folder and save the sound to that folder.
14. Save at least 4 sounds and for images to your App Inventor Folder. (Images can be found in ‘Costumes’)
15. Go back to the App Inventor website. Go to the bottom of the page in the Components window and find the ‘Media’ section. We need to upload the 4 sounds and 4 images to the project so we can use them in the App. Click the “Upload New” button.

16. Navigate to the folder where you stored the images and sounds by clicking ‘Choose File’
17. Select an image or sound and click “Open”

18. Click ‘OK’ to upload the image or sound.
19. You will need to repeat steps 15 to 19 to upload your 4 images and sounds. When you are done you sound see your images and sounds in the Media area:

20. We now need to map each sound to the sound objects. Click on ‘Sound1’ in the Components View:
21. On the Properties window, click ‘Source’ and the list of media will appear. Select a sound to be the source for Sound1 and click ‘OK’:
22. I selected the ‘Dijjeridoo.mp3’ for the sound for Sound1. Now click ‘rename’ in the Components Window and rename Sound1 to match the name of the sound we assigned. Note the use of capital letters and spacing. Object names should have a lowercased starting letter and then use upper case for new words in the middle of the name. This is the convention for computer programming.
23. Assign a sound to each sound object and rename according to the sound using good naming conventions.

All sounds assigned and each Sound object has a unique name.
24. We now need to assign images and names for the buttons. The procedure is similar to the Sound objects. Click on Button1 and set the following properties:

Image: You Choose based on your images
Text: Remove the ‘Text for Button1’ so the text does not appear.

Rename the button based on the image.
25. Assign images and names to the other three buttons. Remember to use good naming conventions.
26. With the Media assigned and the Button and Sound objects named, we are ready to write the program and run this on a phone or emulator. To program the phone we need to use the Blocks Editor. Make sure in this process you read carefully the messages the computer sends.

27. First, click the “Blocks” in the upper right of the User Interface Design page.
28. You will see a screen that looks like this.
29. On the left hand side of the Blocks Editor you will see tabs reading “Built-In”, “My Blocks”, and “Advanced”. Click on the “My Blocks”. You should see the button and sound objects from your app listed on the left.
30. We will now program the first button. Click on one of your buttons (I am going to choose buttonBananas). When you click on the button – you will see a list of available blocks.
31. Drag a (when button.Click) block to the programming area. Remember, the names of your buttons might not match this example.
32. Select one of your sounds from the left and click to reveal its blocks. Drag a ‘sound.Play’ block into the button.Clicked block. Note the names might be different depending on what you named them during the design phase.

33. We now want to test this App on the Phone or Emulator. An Emulator is a virtual ‘phone’ that runs on your computer and allows you to test your App without having an actual Android device on hand. Remember, you need to have installed the App Inventor software for this to work. (Refer to directions at: [http://nebomusic.net/appinventorlessons/install_driver_nexus_phone.pdf](http://nebomusic.net/appinventorlessons/install_driver_nexus_phone.pdf) if you have not installed the software.

34. Find the Icon for ‘aiStarter’ on your desktop. Double click the Icon to start the App Inventor helper program. You will see a black command window open. Do not close this window while using App Inventor. You may minimize the window if needed.
35. Click ‘Connect’ in the App Inventor Menu. Select ‘Emulator’. It might take several minutes for the Emulator to start for the first time.

You will see a message like this:
36. To see the emulator, click on the Android icon located on your TaskBar.
37. After a few minutes, your computer and the emulator will open communications and you should see the App in the Emulator. Sometimes this takes several minutes to start. Be patient . . .

![Emulator screenshot of a drum machine app](image)

38. You should be able to click on one of the buttons (the one you programmed) and it should make the sound. Do a celebratory fist pump if you got this right!!! Do NOT close the emulator. Leave it running while you work so you can continue to test your App.
39. You now need to program the other buttons to play their sounds. Select each button’s .Click event and place the sound.Play block inside the event. The emulator should automatically update to run your App.

![MIT App Inventor 2](image)

40. To complete this assignment you will need to modify the App:

   a. The Drum App needs at least 8 Buttons with Images
      (Hint – you will need to change the Table Arrangement to a 2x4 Table)
      (Hint #2 – change the properties of the Buttons to make them smaller . . .)
      (Hint #3 – Use Small images (Search for 64 by 64 pixel images in Google Images)
   b. The Drum App needs 8 sounds for the images.
   c. The Sounds and Button Images should make sense
      (Have a theme such as animal sounds, movie sounds . . .)

41. Make sure you upload an image of the Code Blocks and the App on the Emulator to your Google Site. No credit for this project will be earned if the content is not on your Google Site.

   (For students with non-functioning Google Sites – See Mr. Michaud for alternative turn in procedures)