Lab –Touch and MultiMedia

Touch and Multimedia

Objectives:

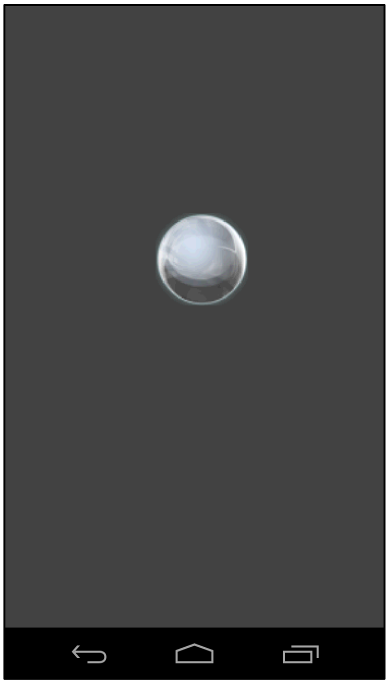
This week’s lab is aimed at getting a better understanding of Touch and Multimedia. Upon completion of this lab you should understand how to have the app respond to touch input, and have it play simple sound effects.

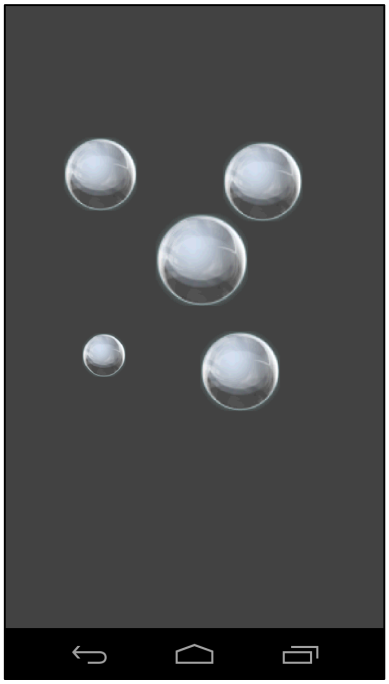
Exercise:

In this part, you will start with skeleton files similar to the end result of last week’s lab.

However, instead of buttons on the menu to add and delete bubbles, you will change this app to add a bubble when the user touches an empty part of the screen.

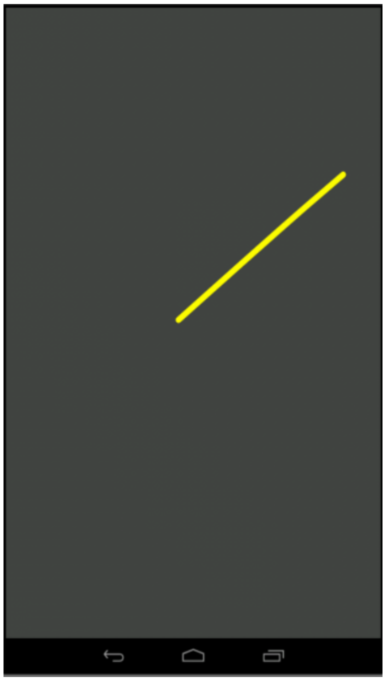
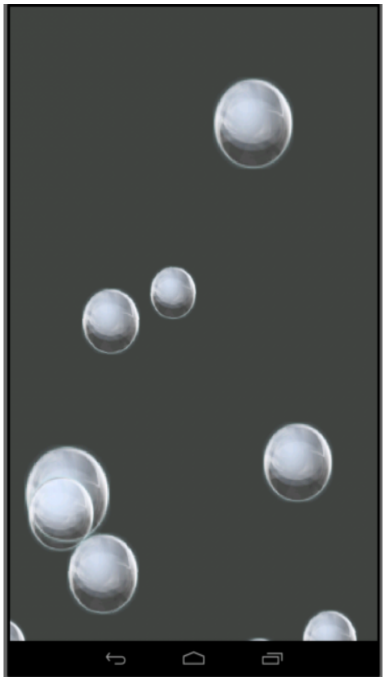
As the user continues to touch empty spaces on the screen, more bubbles should be added.



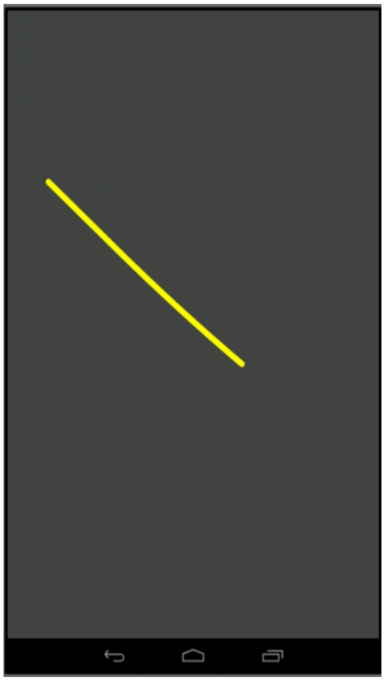
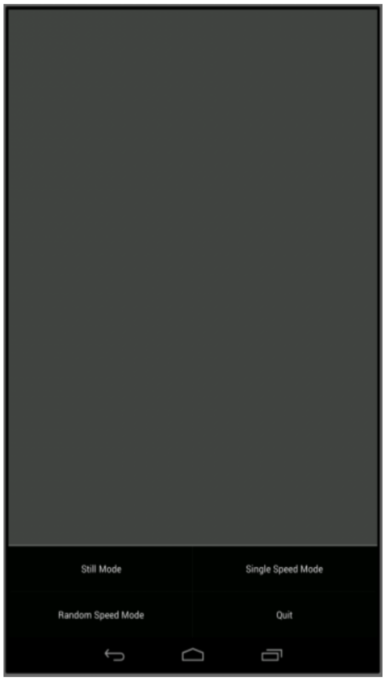


If the user presses the screen at a location already occupied by a bubble, then the bubble should “pop.” That is, it should be removed from the screen and a bubble popping sound should be played. The sound file is in the skeleton code in the /res/raw/ bubble_pop.wav file.

In addition, if the user executes a "fling" gesture starting at a location already occupied on the screen, then the application should change the bubbles current direction and speed to match the direction and speed of the fling gesture.



In addition to the fling gesture, two custom gestures have been defined. A diagonal line from the upper right to the lower left should generate ten random bubbles, and a diagonal line from the upper left to lower right should open the menu.



Implementation Notes:

To obtain the files you need execute the commands we have been using all semester from the root level of your gitlab repository on your local machine:

```
git fetch upstream master
git pull upstream master
git push origin master
```

Once you open the app in Android Studio, look for comments containing the string "TODO" and follow the associated instructions.

Testing:

The test cases for this Lab are in the GesturesLabTest project. You can run the test cases either all at once, by right-clicking the project folder and then selecting Run As > Android JUnit Test, or one at a time, by right-clicking on an individual test case class (e.g., BubbleActivityFling.java) and then continuing as before. The test classes are Robotium test cases.

Warnings:

1. These test cases have been tested on a Galaxy Nexus AVD emulator with API level 26 and 3GB of external storage. To limit configuration problems, you should test your app against a similar AVD.
2. These test cases assume a display screen of at least 550x550 pixels.
3. On some emulators "fling" gestures will sometime report erroneous fling velocities or fail to recognize the fling. Keep an eye out for this erroneous behavior when you're running the BubbleActivityFling test case. The test case should 1) create a new BubbleView, 2) recognize a fling gesture, and 3) remove the BubbleView from mFrame when the BubbleView leaves the screen.

Submission

To submit your work just commit your changes and push to your remote repository on Gitlab.