

# Lab – Graphics and Animation

---

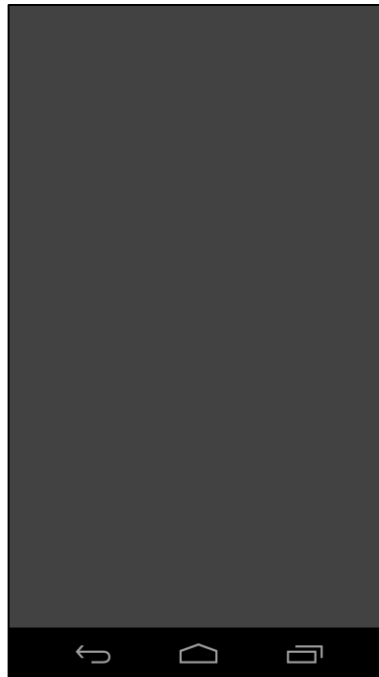
## *Graphics and Animation*

### Objectives:

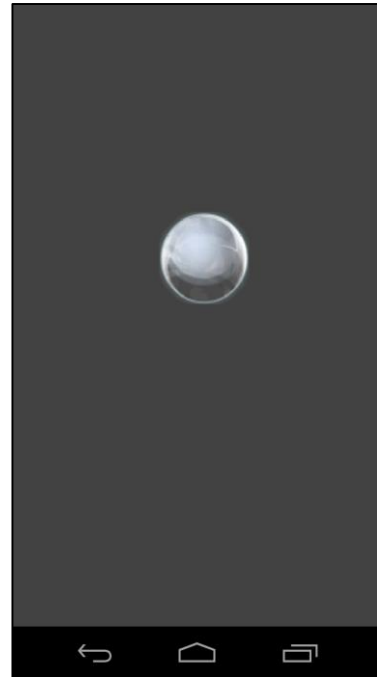
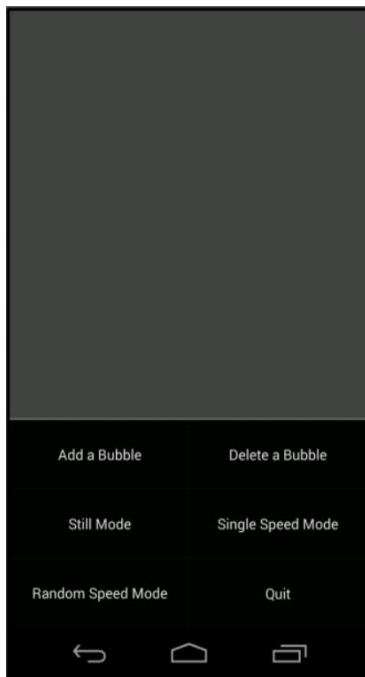
This week's lab is aimed at getting a better understanding of Graphics, Animation, and Multimedia. Upon completion of this lab you should understand how to display and animate images within your application and have it play simple sound effects.

### Exercise:

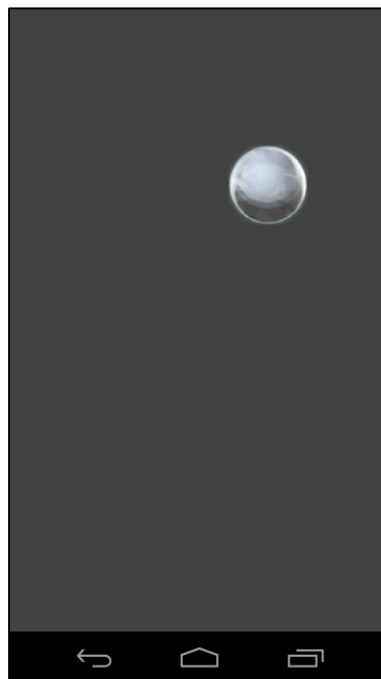
In this part, you will create an application that displays and animates Bubbles. The application's UI will have a main display area that is initially empty as shown below.



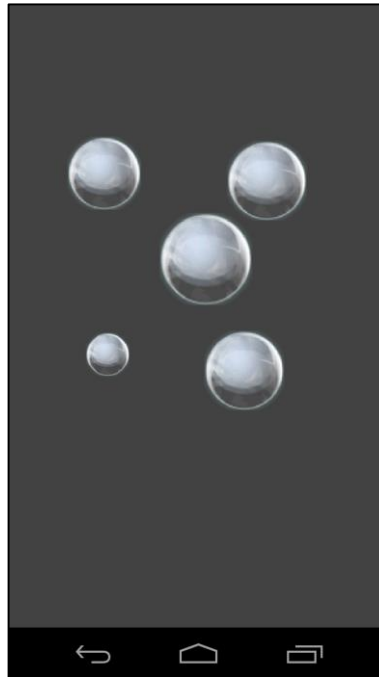
As shown below, when the user presses the back button, the options menu opens. The user then clicks on the “Add a Bubble” menu item and one new bubble should appear on the display. The bubble should then begin to move around the screen. The Bubble's size, direction and speed should be randomized within limits explained in source code skeleton.



Assuming the Bubble's direction is up and to the right, the Bubble shown above might be in the following location after a couple seconds.



As the user continues to press the “Add a Bubble” menu item, more bubbles should be added.



If the user instead presses the “Delete a Bubble” menu item, then the newest bubble that’s still visible 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.

[This screencast shows the app in operation.](#)

### Tips:

Each time your app adds a new Bubble, it should create a new `BubbleView`. The `BubbleView` class handles drawing and moving the Bubble, and also initiates removing the bubble from the main View and playing the popping sound.

New `BubbleViews` must be added to the app's main view, called `mFrame`, otherwise they won't be visible.

You also need to keep track of the Bubbles as they move off the screen. Once a `BubbleView` moves completely off screen, its movement calculations should stop and it should be removed from the app's main View.

When a `BubbleView` is created, its size, movement direction and speed, and rotation speed are randomized, within bounds described in the skeleton code. We have added some special modes to facilitate testing, so we don't expressly test for this behavior in your app. The app will show the menu if the user hits the back button. Click "Quit" in the menu to exit the application.

When a BubbleView changes position, you must notify the system that it has changed, otherwise it will not be redrawn.

## Implementation Notes:

1. Download the application skeleton files and import them into your IDE.
2. Look for comments containing the string "TODO" and follow the associated instructions.

## Testing:

The test cases for this Lab are in the GraphicsLabTest 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 18. 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.

## Submission

To submit your work you will need to submit the GraphicsLab project files we've asked you to modify. These files should be stored in specific directories as described below and then compressed in a zip file. Then you will submit this zip file to the Coursera system. The automatic grading system will test your submission and give you feedback. This process may take some time, especially if many students are submitting at the same time. To make sure your submission is correctly graded, pay attention to the following aspects:

1. Your project files must be compressed in a zip file named GraphicsLab.zip.
2. When decompressed, your submission should contain one top-level directory named GraphicsLabSubmit. Inside that directory there should be one directory named GraphicsLab containing the following file: BubbleActivity.java.