

Lab – Graphics and Animation

Graphics and Animation

Objectives:

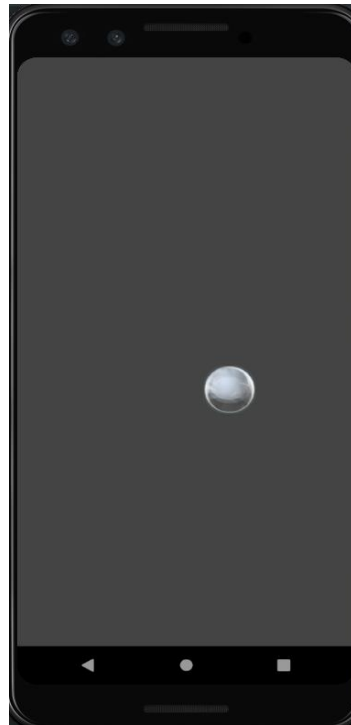
This week's lab is aimed at getting a better understanding of Graphics, Animation, and Multimedia. Upon completion of this lab, you should understand how to display and animate images within your application.

Exercise:

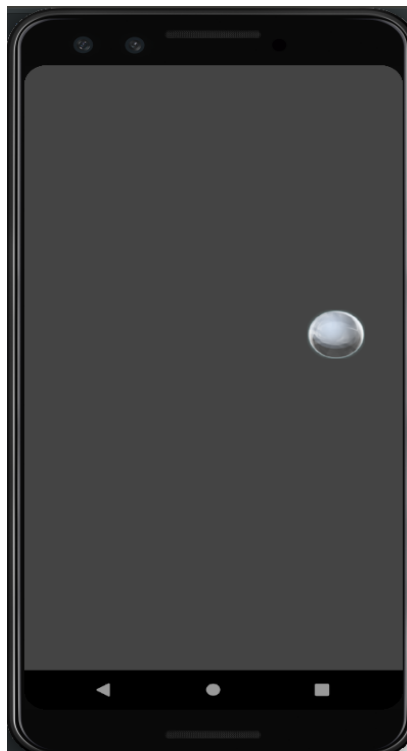
In this part, you will create an application that displays and animates images of Bubbles. The application's UI will have a main display area that is initially empty as shown below.



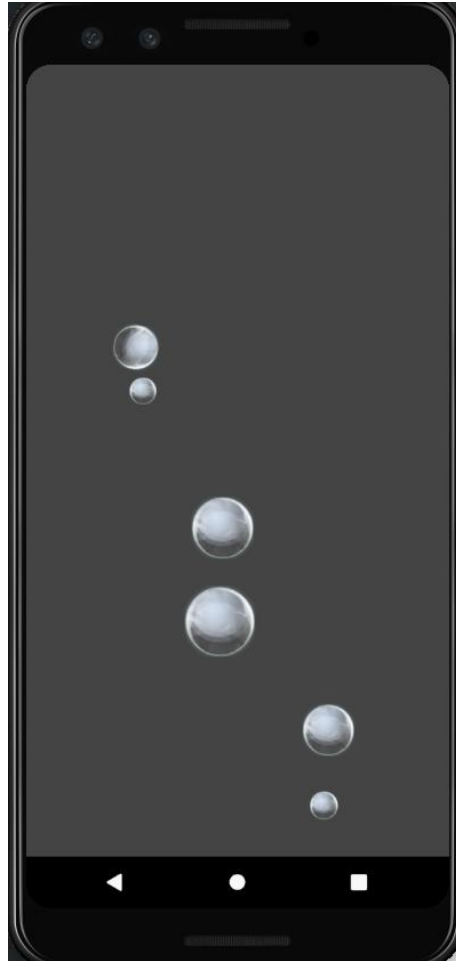
When the user presses the back button, the options menu opens. If the user then clicks on the “Add a Bubble” menu item, one new bubble should appear on the display. The bubble will then begin to move around the screen. The Bubble's size, direction and speed should be randomized within limits explained 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.



If 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 (i.e., the last added bubble that is still visible on the screen) will “pop.” That is, it should be removed from the screen (and a bubble popping sound will be played).

You can find a video record of the expected behavior of the app in operation in the root directory.

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 initiates removing the bubble from the main View.

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.

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

Implementation Notes:

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

Testing:

For this lab we're doing manual tests. You can follow the video record provided in the root directory, execute operations as instructed above and see whether your app behave the same as the video does.

Warnings:

1. These test cases have been tested on a Pixel 5 AVD emulator with API level 31. To limit configuration problems, you should test your app against a similar AVD.

Submission

To submit your work, you will need to commit your solution to your repo on GitLab by running the following command: `git push origin main`.

Note: if you have not already pushed this branch to your repo on GitLab you will need to make a slight modification for this first time and run this instead: `git push -u origin main`. This sets up tracking between your local branch and a branch with the same name on your repo in GitLab.