

Laboratory – Touch

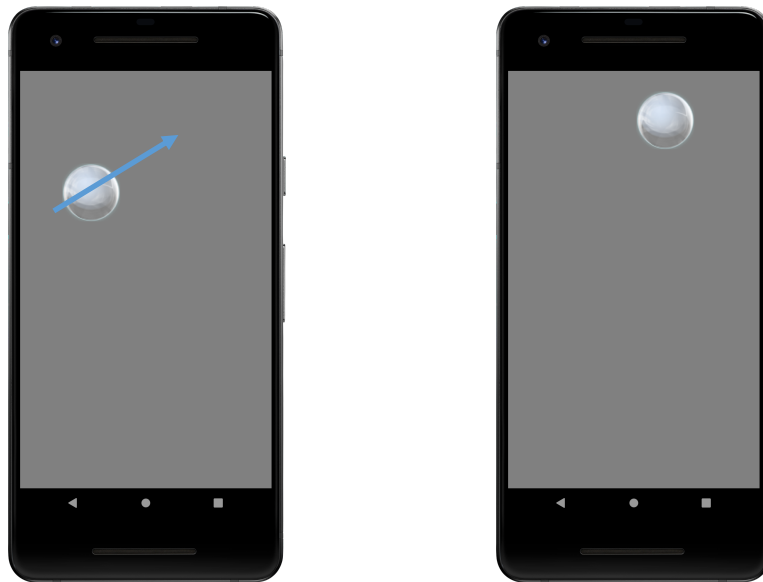
Learn about recognizing and responding to touch gestures

Objectives

Familiarize yourself with creating a gesture detector and attaching it to Views within your application. Create an application that displays the image of a bubble (or ball) that can move in any direction on the display. When the bubble hits the edge of the display it bounces back approximately as one would expect from a real-life object. The app will determine the direction and velocity of the bubble based on user's executing a "fling" gesture on the bubble. Once you've completed this Lab you should understand how to create gesture detectors and how to attach them to specific Views within your app.

LabTouch

This lab involves an app called LabTouch. When it runs, the app displays a user interface like that shown below. Initially, the bubble image drifts across the display, bouncing in a natural way whenever it hits an edge of the display. Over time, the velocity of the bubble's movement will decay until the bubble is no longer moving. The user can modify the movement of the bubble by swiping the bubble with their finger. When this happens the bubble's trajectory is modified, based on the speed and direction of the gesture.



This application includes several components. It includes a ViewModel class called "BubblePositionViewModel," which simulates the movement of the bubble and publishes the current position of the bubble as LiveData. The app also include an Activity called "BubbleActivity." BubbleActivity is responsible for displaying the bubble at its current position. It is also responsible for defining and attaching gesture detecting code to View containing bubble image. It also provides the

BubblePositionViewModel with information about the size of the display and the maximum force of a swiping gesture the device can report and tells the BubblePositionViewModel when to start its simulation. BubblePositionViewModel observes the lifecycle of the BubbleActivity, so that it can pause the simulation when the BubbleActivity is not in the foreground.

See the screencast, LabTouch.mp4, that's included in the Lab directory.

Testing

There are two test cases with several evaluation points. Each evaluation occurs at a step labelled “evaluation point”

The first test case operates as follows:

1. Start the app in portrait mode.
2. Check that the UI shows the bubble (evaluation point 1).
3. Check that the UI shows the bubble hitting an edge of the display and redirecting the bubble’s trajectory (evaluation point 2).

The second test case operates as follows:

1. Start the app in portrait mode.
2. Check that the UI shows the bubble (evaluation point 1)
3. Execute a swiping gesture on the bubble image
4. Check that the bubble’s trajectory is consistent with the direction and velocity of the gesture (evaluation point 2)

After completing your solution, you will record a screencast while performing the manual test. Afterward, you will submit your code and the screencast via git. You can record a screencast using services available in the Logcat console. See: <https://developer.android.com/studio/debug/am-video>

Submission

When you are ready just commit your solution to your repo on GitLab by running the following commands:

```
% git add path/to/changed/files
% git commit -m “completed Lab10_Touch”
% 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.

Some Implementation Notes

We are providing template code and layout resources for this application. Only modify the areas marked with the word `TODO`.

We have done our testing on an emulator using a Pixel 5 AVD with API level 31. To limit configuration problems, you should test your app against a similar AVD.