CS194A

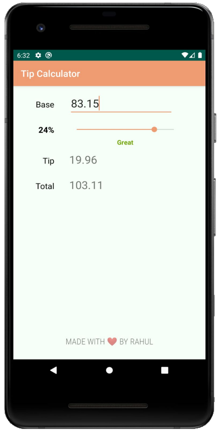
Android Programming Workshop

Lecture 9: November 11, 2020

Rahul Pandey

Outline

- Logistics
- Fragments
- Services
- Android testing
- App architecture
- Mobile gaming
- Alternatives to native app development



6:32

Tip Calculator

Base 83.15

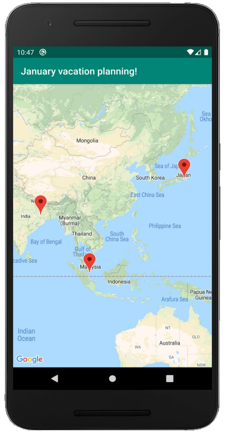
24%

Great

Tip 19.96

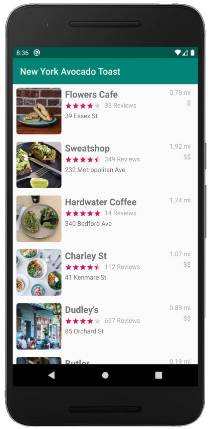
Total 103.11

MADE WITH ❤ BY RAHUL



10:47

January vacation planning!



8:36

New York Avocado Toast

Flowers Cafe 0.78 mi
38 Reviews $
39 Essex St

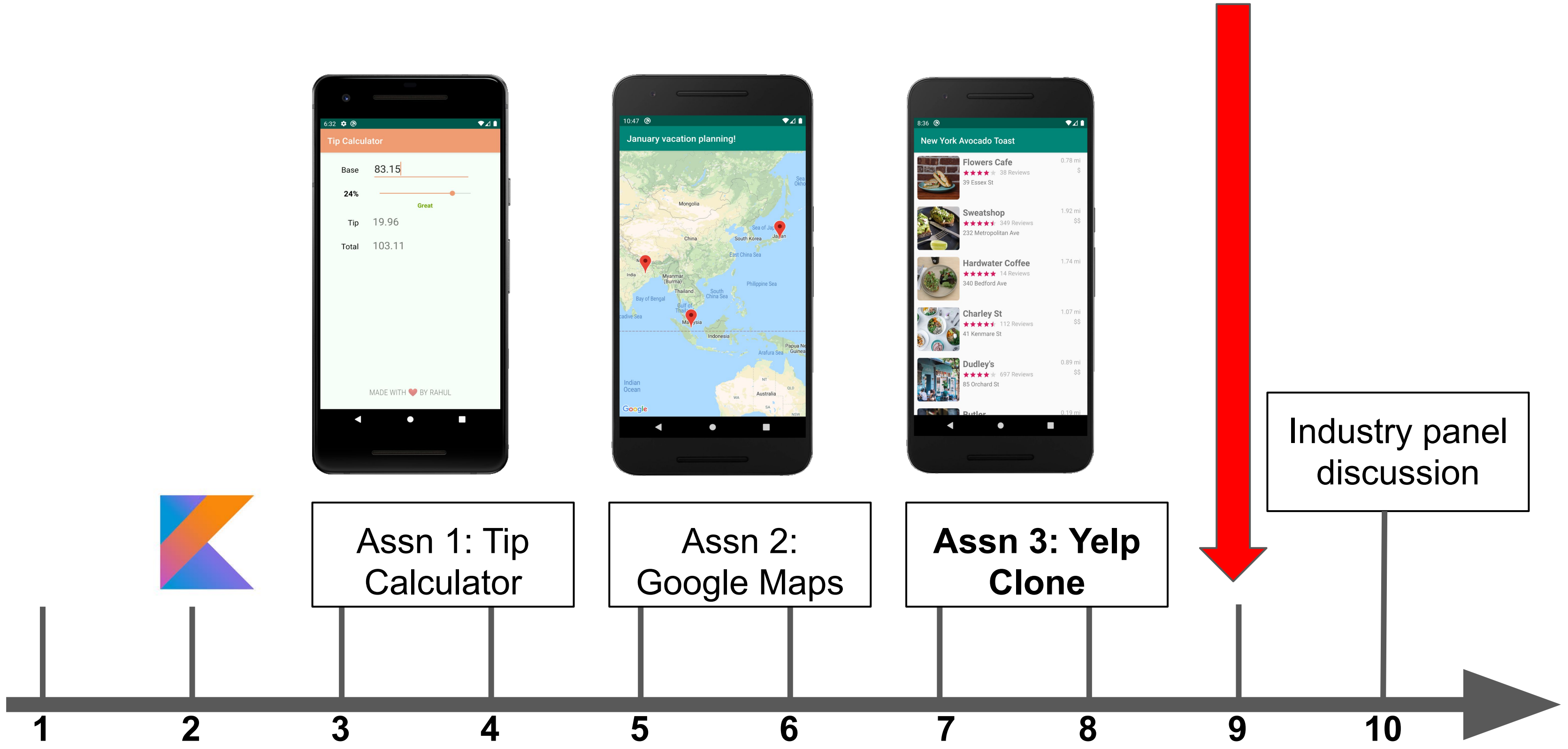
Sweatshop 1.92 mi
349 Reviews $$
232 Metropolitan Ave

Hardwater Coffee 1.74 mi
14 Reviews
340 Bedford Ave

Charley St 1.07 mi
112 Reviews $$
41 Kenmare St

Dudley's 0.89 mi
697 Reviews $$
85 Orchard St

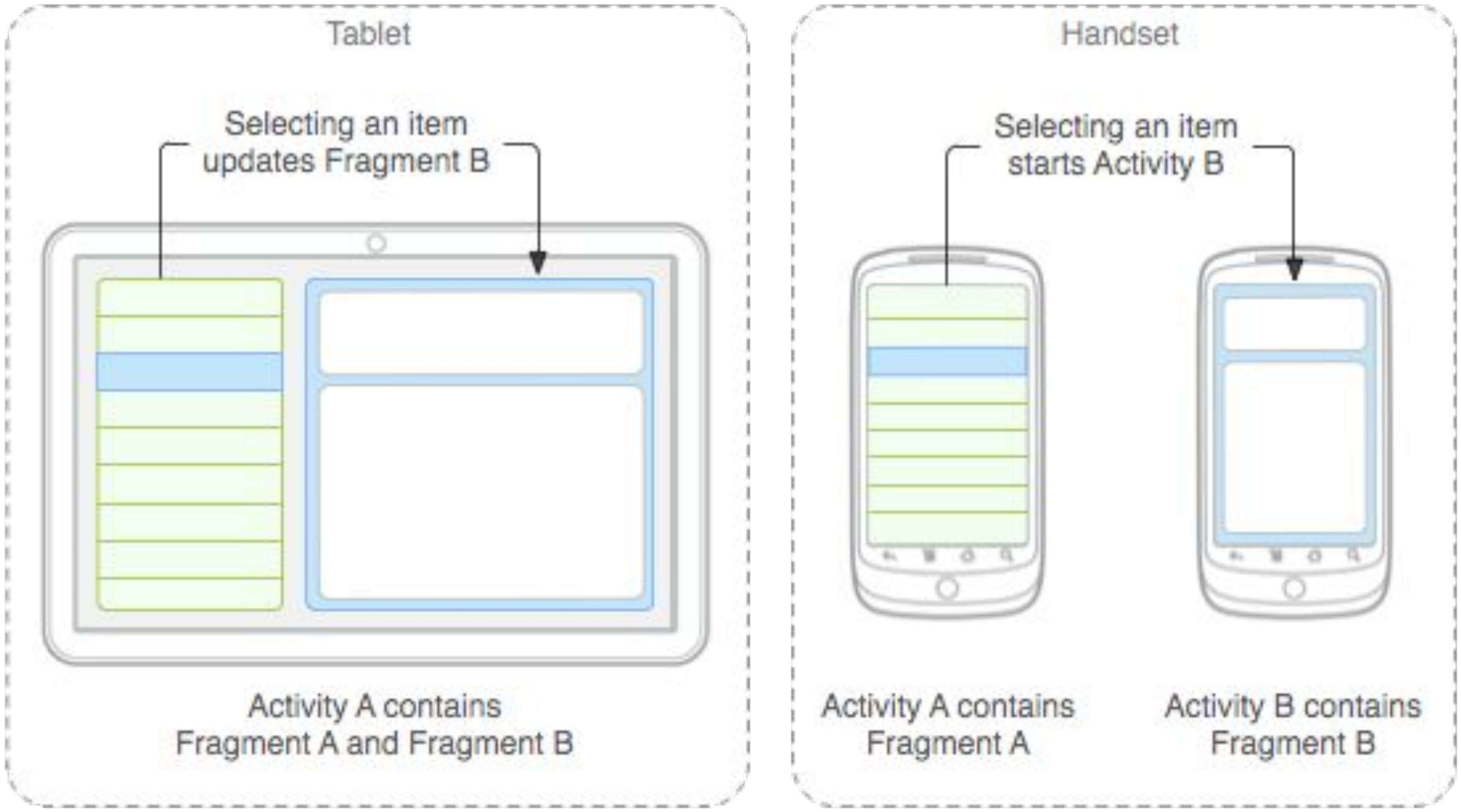
Butler 0.19 mi



Week #

Fragments

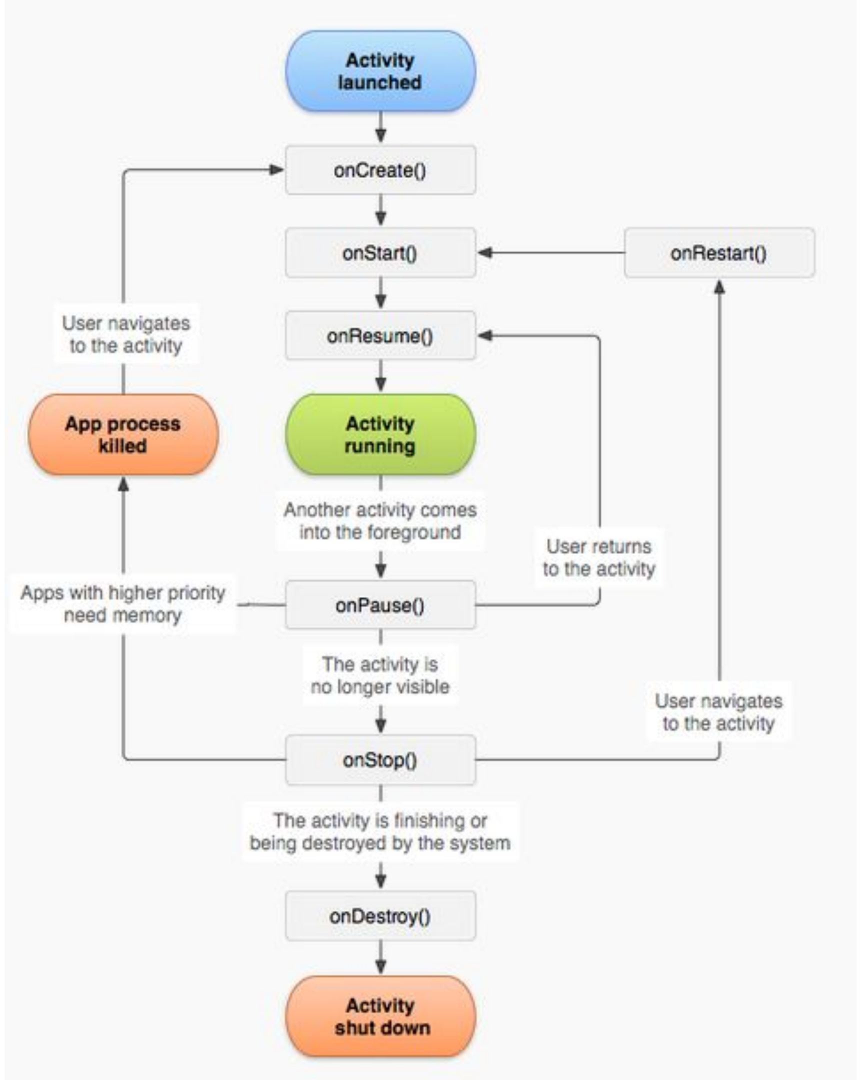
- A reusable portion of UI that lives inside an Activity.
- Multiple fragments can be combined in one activity
  - Helps to handle different devices and screen sizes
  - Helps to reuse common UI across your app
- Has its own lifecycle similar to the Activity lifecycle


Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B

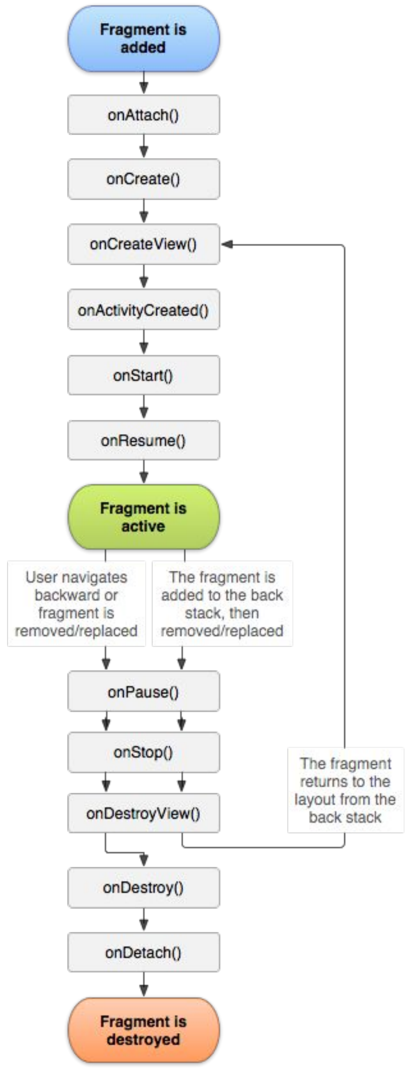
Fragment vs Activity

- Methods defined on the activity are not available in the fragment
    - Need to use the activity property to access the enclosing activity
- Passing/accessing information in a fragment (intents/bundles) is done by asking the enclosing activity
- Fragment initialization and lifecycle are different
    - Activity: `onCreate`
    - Fragment: `onActivityCreated`

Activity lifecycle


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

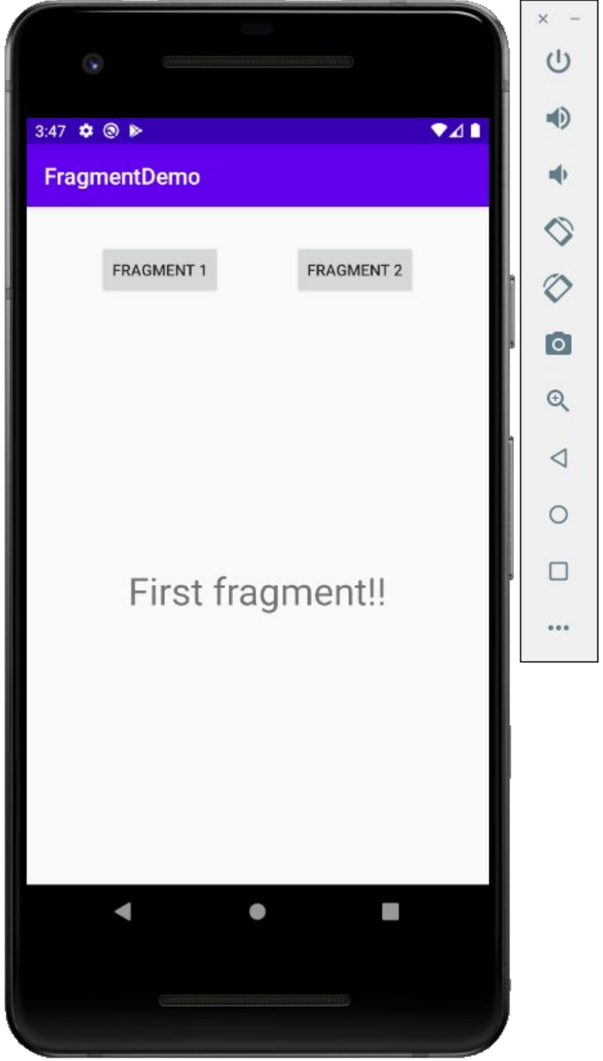
Fragment lifecycle



Using fragments

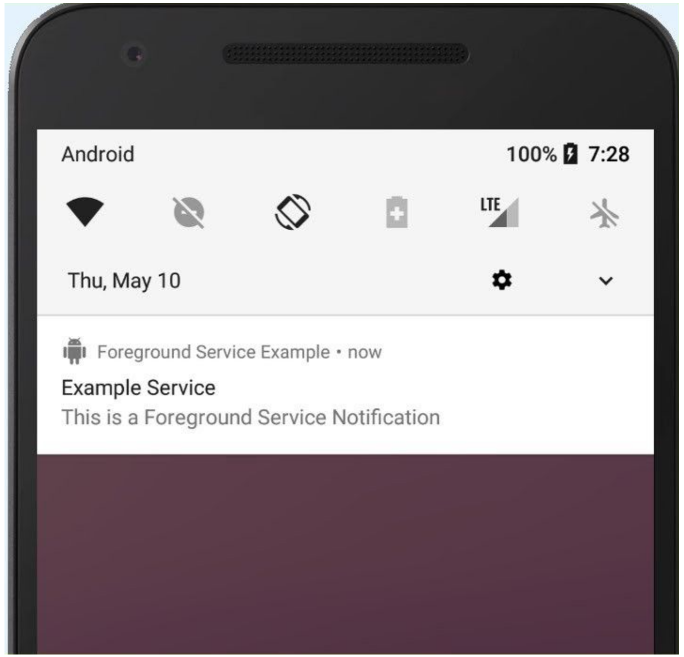
- Static fragment
  - Add the `fragment` component in the activity
- Dynamic fragment
  - Add container to the layout and programmatically add the fragment

Code pointer



Services

- A background task used by an app
  - Use to perform long-running operations without a UI
  - Examples: handle network transactions, play music, perform file I/O
- Also has a lifecycle
  - onCreate, onStartCommand, onDestroy
- Can broadcast a result when a task is completed
  - Applications can hear broadcasts using a BroadcastReceiver

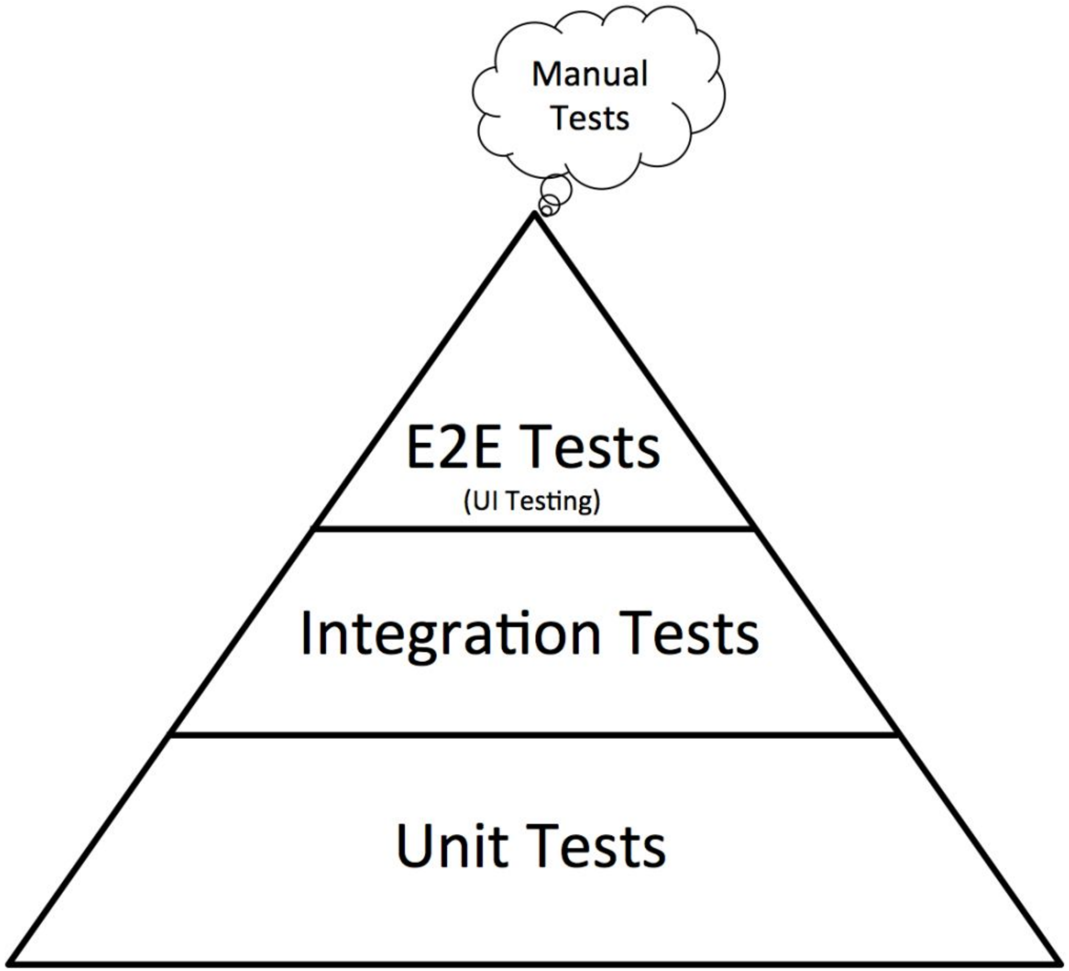

Android 100% 7:28

Thu, May 10

Foreground Service Example • now

Example Service

This is a Foreground Service Notification


Manual Tests
E2E Tests
(UI Testing)
Integration Tests
Unit Tests

Source

Android testing

Why is it hard?

- **End to end testing**: directly testing the UI is flaky, e.g. dropdown takes time to render
  - Espresso: requires an emulator
- **Integration testing**: check interaction between different components
  - Robolectric: mock Android components
- **Unit testing**: just one component, no interaction with Android framework
  - JUnit/Mockito: very fast to run

App architecture

- Architecture: “fundamental structures of a software system”
- Architecture educates how you organize your code
    - Has consequences around how easy it is to debug your code or onboard new devs
- Different components of all mobile apps:
    - UI or view layer
    - Data classes or models
    - Repository: holds or retrieves the data
    - “Business logic” component for responding to user input

Android app architectures

- **Objective**: avoid having all logic live inside Activities

- **MVC**: Model View Controller
- **MVP**: Model View Presenter
- **MVVM**: Model View ViewModel

Gaming

Why use game engines?

- Cross platform capability
- Handles complexities around physics, lighting, 2D/3D graphics
- Games typically don’t have as many platform-specific differences

unity

Alternatives to native app development

What problem are they trying to solve?

1. Double the effort to ship an app two platforms (Android + iOS)
2. Developer experience sucks- gatekeepers to releasing your code

React Native

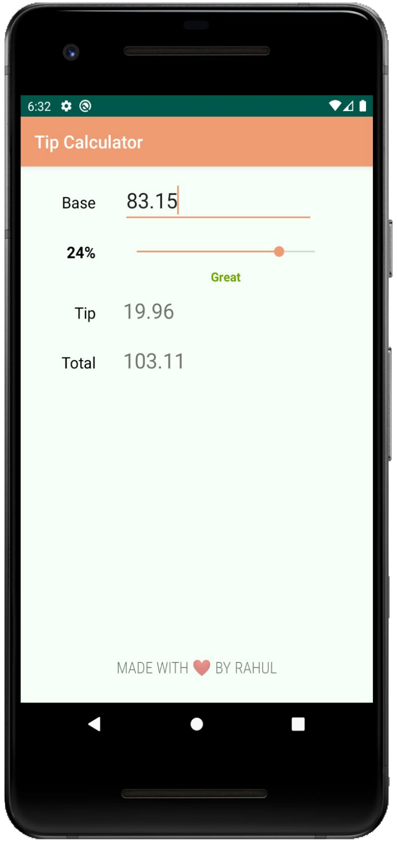
- Developed by Facebook for cross platform app development (move faster)
- Uses JavaScript, source code is converted to native elements
- Tight platform integration, ability to write modules in platform code
- Heavily used across industry: Facebook, Uber, Walmart

Flutter

- Created by Google
- UI toolkit for building native apps for mobile, web, and desktop in a single codebase
- Provides a full native experience by using native compilers
- Uses the Dart programming language
- Less mature than React Native but growing

Xamarin

- Bought by Microsoft
- Uses .NET and C# (Microsoft technologies)
- Also turns into native code
- 75% code sharing between platforms
- Not as popular in Silicon Valley, but has a mature community


6:32

Tip Calculator

Base 83.15

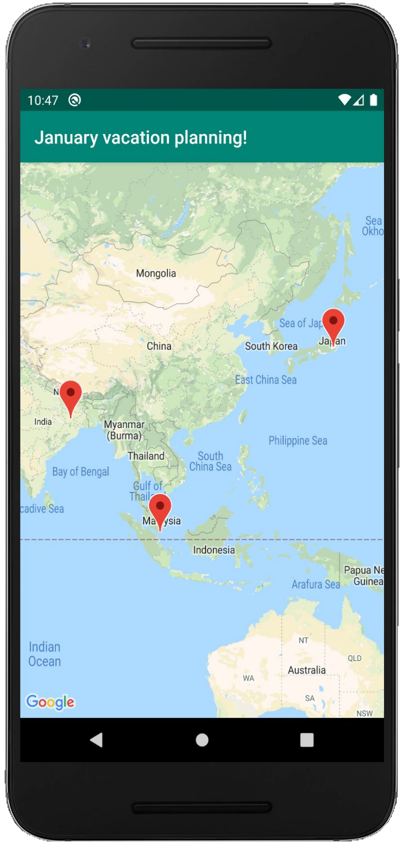
24%

Great

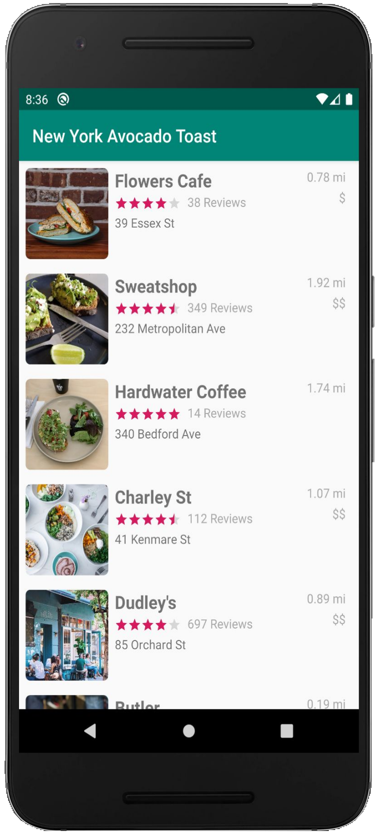
Tip 19.96

Total 103.11

MADE WITH ❤ BY RAHUL


10:47

January vacation planning!


8:36

New York Avocado Toast

Flowers Cafe	★★★★☆ 38 Reviews	0.78 mi
39 Essex St		$
Sweatshop	★★★★½ 349 Reviews	1.92 mi
232 Metropolitan Ave		$$
Hardwater Coffee	★★★★★ 14 Reviews	1.74 mi
340 Bedford Ave		
Charley St	★★★★½ 112 Reviews	1.07 mi
41 Kenmare St		$$
Dudley's	★★★★☆ 697 Reviews	0.89 mi
85 Orchard St		$$
Butler		0.19 mi

If you want to publish your app...

- Create a Google developer account
- Create assets for publishing: icon, screenshots, description
- Use a unique package name
- Create a keystore in Android Studio and singing key for the APK
- (Optional) Use Proguard to minify your app
- (Optional) Add an analytics/crash tracking library
- Publish + profit!