Lesson 8:

Cupcake Order

Instructor: Ahmet Geymen

About this lesson

- Lesson 8:
  - Unit Tests
  - Navigation
- Workshop
  - Unit Tests (Unscrambler)
  - Navigation (Cupcake)

Get started

Unit Tests

Advantages of tests

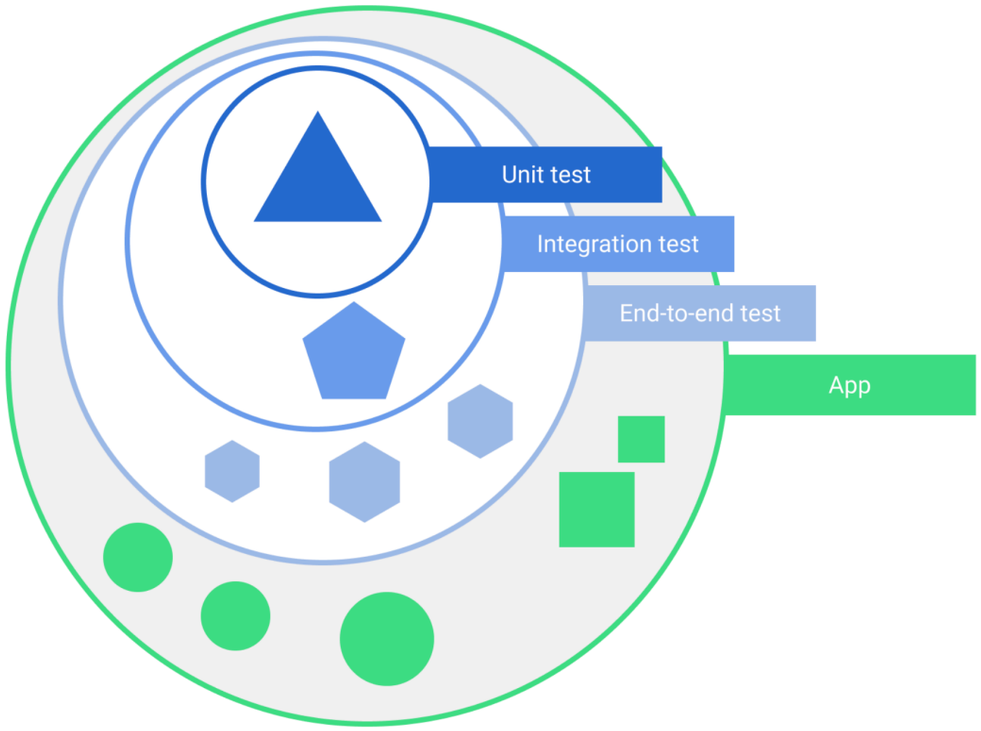
- Rapid feedback on failures.
- Early failure detection in the development cycle.
- Safer code refactoring, allowing you to optimize code without worrying about regressions.
- Stable development velocity, helping you minimize technical debt.

Types of tests

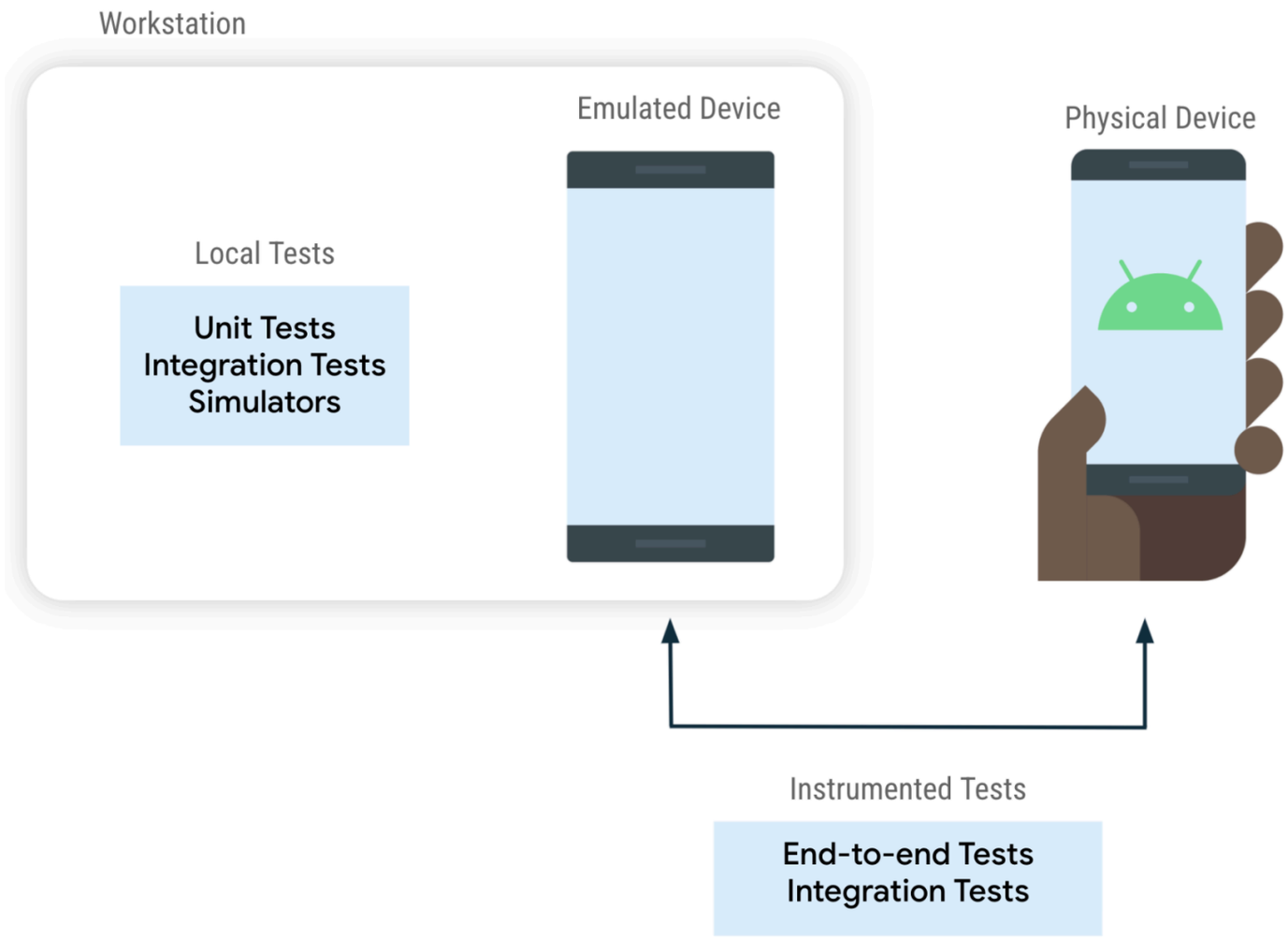
- Depending on subject:
  - Functional testing: does my app do what it's supposed to?
  - Performance testing: does it do it quickly and efficiently?
  - Accessibility testing: does it work well with accessibility services?
  - Compatibility testing: does it work well on every device and API level?

Types of tests

- Depending on scope:
  - Unit tests or small tests only verify a very small portion of the app, such as a method or class.
  - Medium tests are in between and check the integration between two or more units
  - End-to-end tests or big tests verify larger parts of the app at the same time, such as a whole screen or user flow.



Types of tests



Properties of unit tests

- Focused: It should focus on testing a unit, such as a piece of code. This piece of code is often a class or a method.

- Understandable: It should be simple and easy to understand when you read the code. At a glance, a developer should be able to immediately understand the intention behind the test.

- Deterministic: It should consistently pass or fail. When you run the tests any number of times, without making any code changes, the test should yield the same result.

- Self-contained: It does not require any human interaction or setup and runs in isolation.

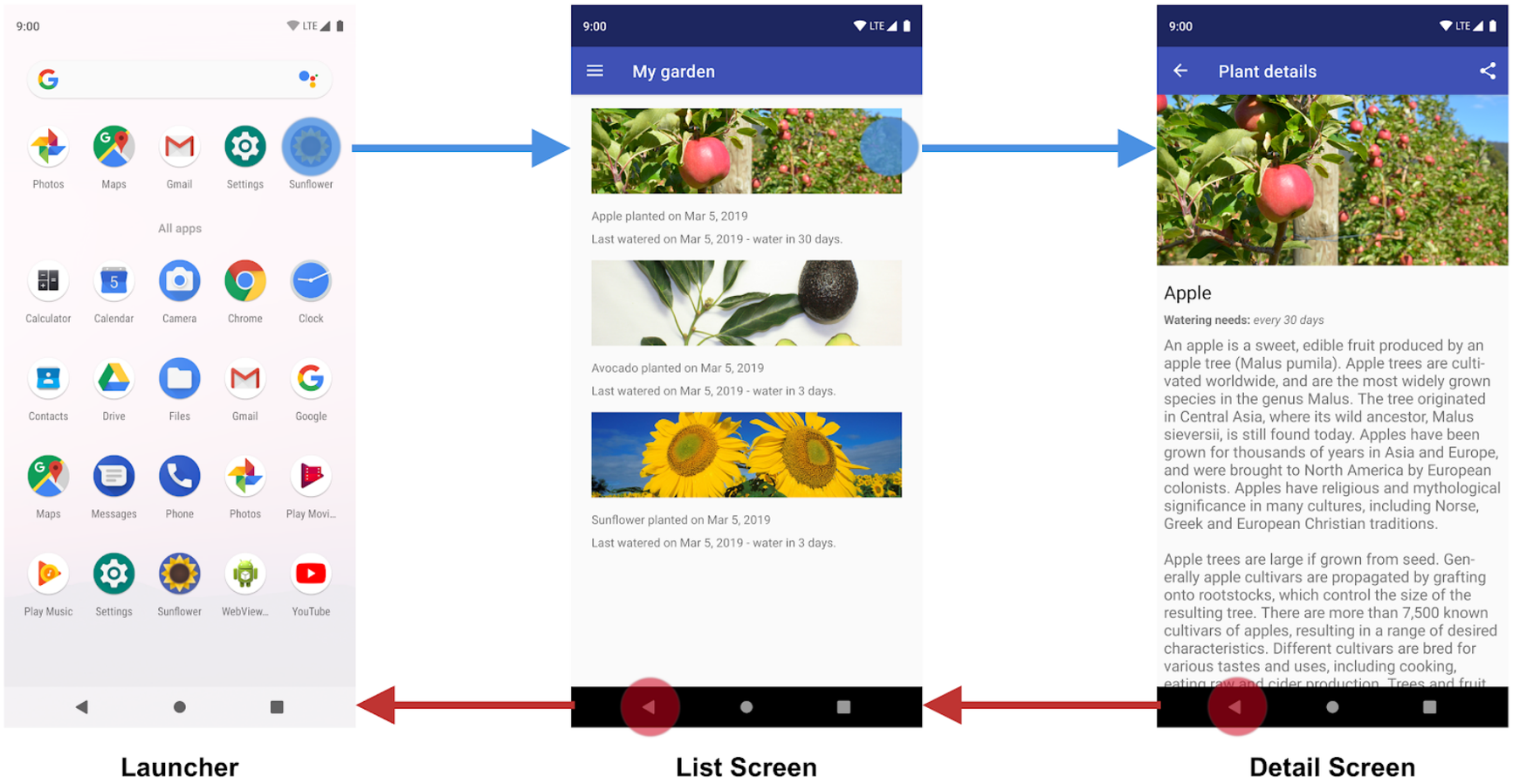
Test strategy

- Success path:
  - Also known as happy path tests, focus on testing the functionality for a positive flow.
- Error path:
  - The error path tests focus on testing the functionality for a negative flow.
- Boundary case:
  - Boundary case focuses on testing boundary conditions in the app.

Navigation

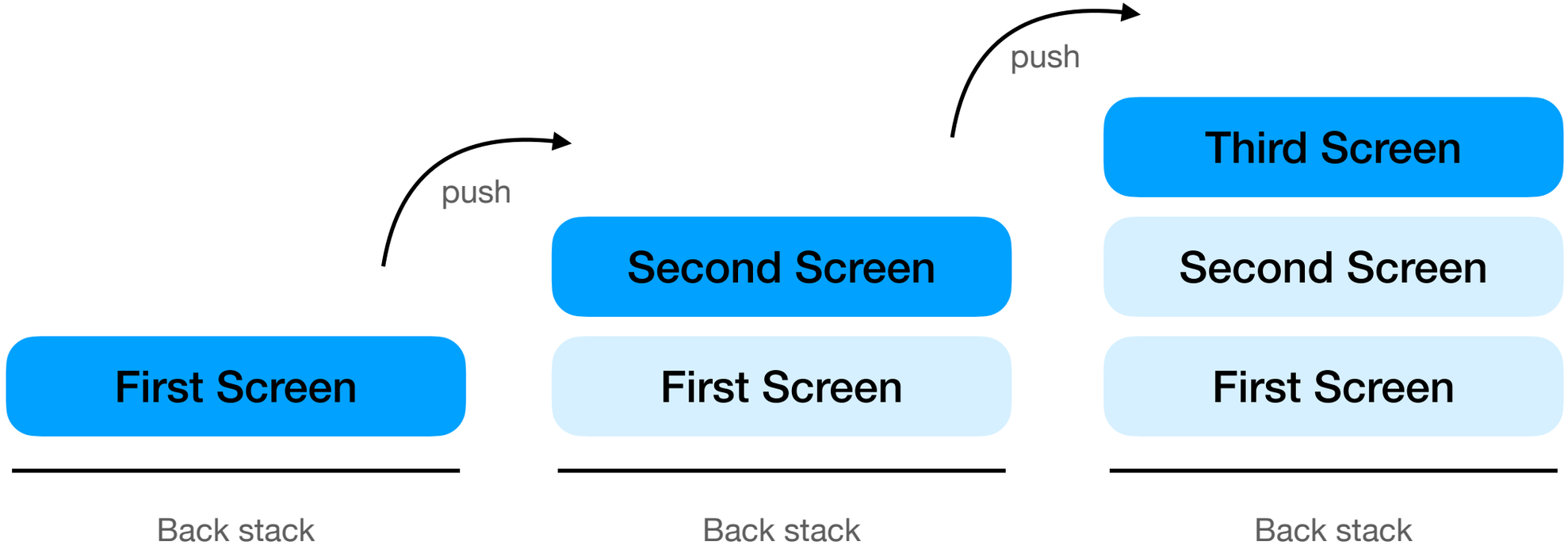
Principles of navigation

- Fixed start destination



Principles of navigation

- Navigation state is represented as a stack of destinations
- Start destination is base of the back stack



Navigation components

- Destination(s)
- NavHost
- NavController
- Route

```
NavHost(
        navController,
        startDestination,
        modifier,
) {
        content
}
```

Navigation components

- Define NavGraph with extension function composable()

```
composable( route ) {
    content
}
```

- Navigate trough navigation controller using route IDs

```
navController.navigate( route )
```

Workshop