UI Testing

Upon completion of this module, a student will be able to

- understand and explain the purpose of UI Testing
- configure an app to be tested with espresso
- setup to test an activity
- create an espresso test
- create a custom matcher tor testing

Assignment

- Task
  - Add and test a UI to your calculator
- Repo
  - https://github.com/LambdaSchool/AndroidUITesting
- Submission
  - Fork on github and submit pull request

A Student Can

understand and explain the purpose of UI Testing

User Interface Testing

- Run on Android OS
- Can see tests being run on device

A Student Can

configure an app to be tested with espresso

Preparing to test UI with Espresso

- Disable animations in Developer Settings
- Add dependency to gradle

A Student Can

setup to test an activity

Test Setup

- New test in androidTest
- @Rule
  - new ActivityTestRule<>(/* Activity under test */)
- @Before

A Student Can

create an espresso test

Get a Handle to a UI

- onView(withId())

Perform Action in UI

- perform() - ViewInteraction
  - click() - ViewActions
  - typeText(String)

Check Result

- check() - ViewInteraction
  - matches() – ViewAssertions
    - withText() – ViewMatchers
    - isChecked()
    - isDisplayed()

A Student Can

create a custom matcher tor testing

Custom Matcher

- https://medium.com/@dbottillo/android-ui-test-espresso-matcher-for-imageview-1a28c832626f