Testing Summary

Android Integration Tests

at androidTest/com.olfybsppa.inglesaventurero.tests/xxxxx

1. scripttests has integration tests. I made up a scene (find at java/stageactivity/AlphabeticalTracker.java and at AlphabetTracker-java.odt) and traverse it up and down using the tests under scripttests. Also see integration-tests.txt to see how framework is set up to run these tests.

2. resolvertests test simple and corner cases for the resolvers (AttributionsResolver, SceneResolver, ResolverWrapper....) Simple cases are insertions and deletions. Corner cases is adding duplicate data, or updates.

3. collectortests test the following:
    1. that collector objects (hints, pages, attributions...) can be converted into ContentValues or Parcelables and back again. Note, Android does not allow testing ContentValues and Parcelables in UnitTests, so they are tested here.

4. linesCPtests assert that the rules for entering and deleting a collector object into LinesCP, which extends ContentProvider, are followed. (Collector Objects are Hint, Reply, Background, ImageCredit...) All collector objects have tests for insert, delete, and their table rules such as 'unique' and 'not null'. ResolverWrapper only has methods for updateBackground and updateImageCredit, so those are the only update() methods that are tested.
    1. ProviderInsertHintsTest:
       a. not testing bulk insert for Hint test. Code is simple enough in LinesCP.
    2. TODO: RepliesTest should test scene_id can not be null.
    3. TODO: Inserting isn't always tested by retrieving object and testing if it equals the inserted object. Instead just the count is tested. Is done properly for Hint and Reply.
    4. TODO: Scene doesn't test english_title must be unique and spanish_title must be unique.

5. utilsTests tests classes that are basic utilities. Included is Ez.convertToBundle() because Android doesn't allow Bundle in Unit tests. Also included is URIDeterminerTest.

6.

Unit Tests

at test/java/com.olfybsppa.inglesaventurero.tests/xxxxx

1. instruction_prompter_opening_activity:
  Opening Activity shows the user some helpful prompts when they first start using app. These are triggered when Opening Activity finds out how many rows it has. But it usually finds out on a background thread, so DelayedPrompter, a Handler, is given the information and notifies OpeningAcitivity when the user thread is available.
    1. In DelayedPrompterUTest, I'm spying on DelayedPrompter to make sure it notifies it's listeners when it receives a handleMessage with the NUM_OF_ROWS_IN_LIST.
    2. PromptCoordinator tests different scenarios where num of rows in list varies and the user has seen varying instructions. The PromptCoordinator always sends the correct DailogFragment which has instructions for the user.

2. stagehouseactivityutest/collectors_basics:
For Hint, Reply, and Page these are the basic getter, setter, equals, comparable tests.

3. stagehouseactivityutest/collectors_interactions:
Pages hold Hints, Replies, and ReplyLineSets. Replies and ReplyLineSets implement Matchable.

1. PageMatchUTest: The Page method acceptResponses(responses) returns an Answer which holds basic info, like whether the page has Matchables, was it matched, was it already matched... These tests make sure that when acceptResponses is called, that the correct Answer is returned.
2. PageGetAptHintUTest: A page has a combination of matched and unmatched Matchables, and hints in different orders. These test make sure that the correct next hint is returned using Page.getAptLeader();
3. PageGetFollowingPage_UTest
4. ReplyLimitedUTest: Tests that reply will match a list of responses properly. TODO: Only testing positive match case, should also check not-a-match case.

4. stagehouseactivityutest/tracker:
Every story has many paths depending on what the user chooses. Tracker keeps track of the current story.

5. webscenelistactivityutests/collectors_basics:
Basic getters, setters, for ImageCredit, CPHint, CPReply, CPPage, CPScene and that CPHint, CPReply, and ImageCredit can be extracted from a mock cursor. *TODO: CPPageTest is not done??*

6. webscenelistactivityutests/ (all resolvers):
Essentially tests all resolvers by spying on them. AttributionsResolver, ImageCreditResolver, PageResolver, SceneResolver all hold an instance of ResolverWrapper. ResolverWrapper contains a ContentResolver. Tests that AttributionsResolver, ICResolver, PageResolver, SceneResolver all call the correct methods on each other when necessary, and also on ResolverWrapper. There are some interesting corner cases, for instance, AttributionResolver delete must call clear() methods on ImageCreditResolver, or not depending if there are any Attributions still pointing to that ImageCredit. When AttributionResolver sets new Attributions to a Background, the old attibutions that are not in the new list need to be deleted, and AttributionResolver is spied on to make sure that ResolverWrapper delete methods are called.
So all these spy tests on the resolvers are unnecessary because the resolvers are thoroughly tested in the integration tests above.

7. webscenelistactivityutests/DocToSceneWorkerUTests:
Tests DocToSceneWorker.work(), which takes a document and returns a Scene, Pages, Backgrounds, Hints, Replies, and Attributions. The document it uses is in resources/alphabet_doc_1.txt.

Manual Testing

at InglesAventurero/AndroidStudio/Testing/testing-manual-tests.odt