

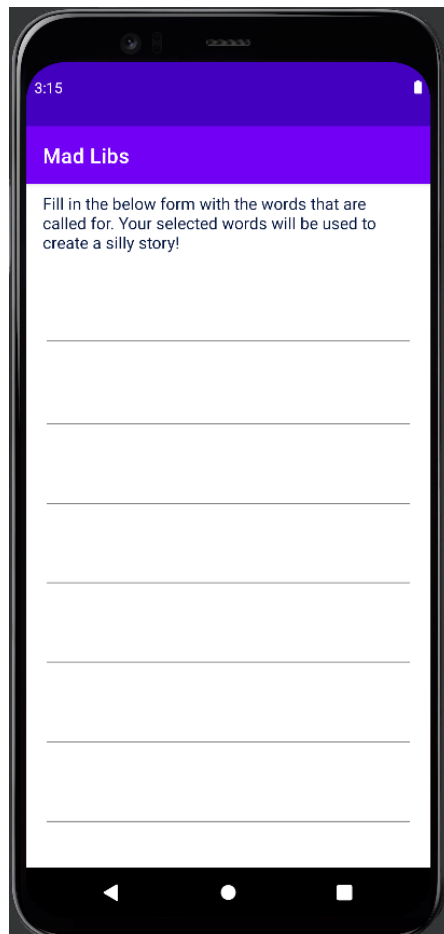

MAD LIBS LAB

In this lab, you will modify the layout for a mad libs game. The player will be able to select different types of words which will be used to create a silly story!

PART 1: OPENING THE PROJECT

Open the provided project template and run the app (clicking on the “Run app” icon  or “Debug app” icon  located in the menu near the top right hand side of the Android Studio window).

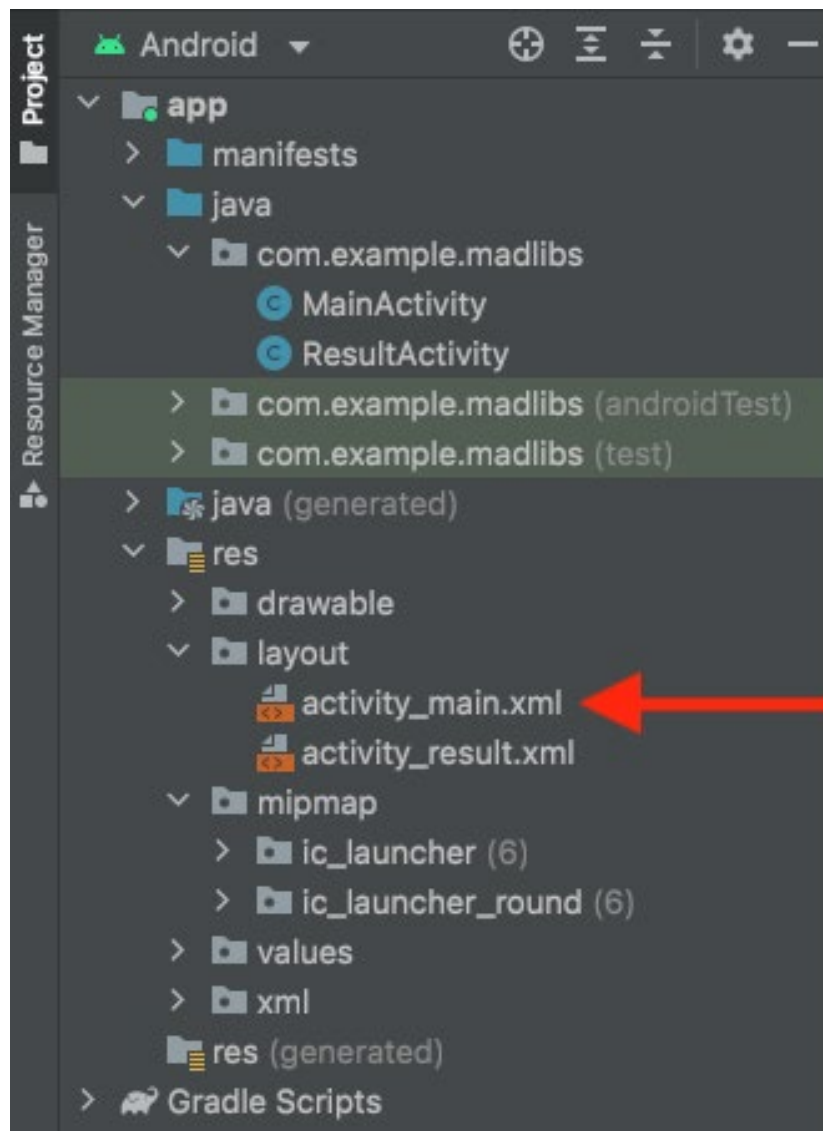
You will see the Mad Libs app successfully launch on the emulator with a description about how the game works followed by several blank lines and a “DONE” button. There is no indication as to what kind of word should be chosen to fill in each blank, and clicking on the “DONE” button does not seem to affect anything in the app unless all blanks are filled.



PART 2: MODIFYING THE LAYOUT

To address the issues mentioned above, you'll be making some modifications to the layout of the app.

Click on the “Project” tab on the left side of the Android Studio window to open up the Android project structure, then open the layout file “activity_main.xml” under app/res/layout folder.



The layout of the Mad Libs app you just looked through is defined here in “activity_main.xml”. For each word that the game calls for, there is:

- **An EditText with ID “[word type]_edittext”** – the blank line in which the player will fill in the word

- **A TextView with ID “[word type]_errmsg”** – the error message shown should the player leave the **EditText** mentioned above empty

Overall, the order of words to fill all the blanks in the game is:

1. Foreign country
2. Adverb
3. Adverb
4. Part of the body
5. Verb
6. Adjective
7. Noun
8. Verb
9. Adjective
10. Part of the body
11. Verb

Task #1: Let the player know what kind of word to fill in each blank

Here are a few ways to accomplish this:

1. Using the “hint” attribute in the EditText to specify the type of word

The “hint” attribute should “hint” to the player what to fill an EditText with. See the example below.

```
<EditText
    android:id="@+id/foreign_country_edittext"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Foreign country"/>
```

In the app, adding a hint to an EditText should look something like this:

Foreign country

Adverb

Adverb

2. Adding a TextView above the word's EditText to specify the type of word

The extra label placed before the EditText can also indicate to the player what word to fill into the EditText.

```
<TextView
    android:id="@+id/foreign_country_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="18sp"
    android:text="Foreign country"/>

<EditText
    android:id="@+id/foreign_country_edittext"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

Adding the TextView should look something like this in the app:

Foreign country

Adverb

Adverb

3. Using the “hint” attribute in the EditText to inspire

You can also label the EditText with a TextView, and use the EditText’s “hint” attribute to give the player some ideas as to what word to fill in the blank.

Foreign country

Think of a land far far away...

Adverb

Word that describes a verb

Adverb

Describes a verb, usually ends in '-ly'

Feel free to experiment with different ways of doing this task, and consider which would make for the best player experience.

Task #2: Let the player know they must fill in every blank

Since clicking on the “DONE” button prevents the player from moving on in the game if they didn’t fill in all the blanks with words, the player needs to know what they are doing incorrectly.

Currently, each word’s error message TextView is not visible on the app because there is no message set in the “text” attribute of the TextView. If not specified, the “text” attribute of any TextView is set to an empty string (“”) by default.

For each word’s error message TextView, set the “text” attribute of that TextView to an error message telling the player what kind of word they should fill the blank with.

Refer to the example below.

```
<TextView
    android:id="@+id/foreign_country_errmsg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:textSize="12sp"
    android:textColor="#FFCC0000"
    android:text="Please enter the name of a foreign country!"/>
```

After setting the “text” attribute for each word’s error message TextView, all error messages are now visible right when the app is launched:

Mad Libs

Fill in the below form with the words that are called for. Your selected words will be used to create a silly story!

Foreign country

Think of a land far far away...

Please enter the name of a foreign country!

Adverb

Word that describes a verb

Please enter an adverb!

Adverb

Describes a verb, usually ends in '-ly'

Please enter an adverb!

You can, of course, leave it this way, but there is really no need to see any error messages when no error has been made on the player’s part yet. The player only needs to be warned about leaving a blank empty when that error is present by the time they press on the “DONE” button.

One thing that the code in “MainActivity.java” (under app/java/com/example/madlibs folder) takes care of is displaying an error message when the “DONE” button is pressed only when the associated blank is found to not have been filled. This is done by setting the “visibility” attribute of the error message TextView to “visible”.

If not specified, the “visibility” attribute of any TextView is set to “visible” by default. **To avoid displaying the error messages right when the Mad Libs game is launched, set the “visibility” attribute of the error message TextViews to “invisible”.**

```
<TextView
    android:id="@+id/foreign_country_errmsg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:textSize="12sp"
    android:textColor="#FFCC0000"
    android:text="Please enter the name of a foreign country!"
    android:visibility="invisible"/>
```

Now when launching the app, the error messages should initially be invisible. Error messages should appear only if the player didn't have all of the blanks filled by the time they are submitting their words for the game.

Foreign country

Think of a land far far away...

Please enter the name of a foreign country!

Adverb

happily

Adverb

Describes a verb, usually ends in '-ly'

Please enter an adverb!

PART 3: TESTING THE APP

When playing the game, you should be able to know which word to fill each blank with, and if you leave any of them empty, pressing the “DONE” button should reveal an error message telling you to fill the associated blank.

Play through the game, and after filling all the blanks, you should generate a passage about walking on glass that may or not make sense!

