CS 591 – Mobile Application Development

Fragments and App Development Practice

Date: _____________

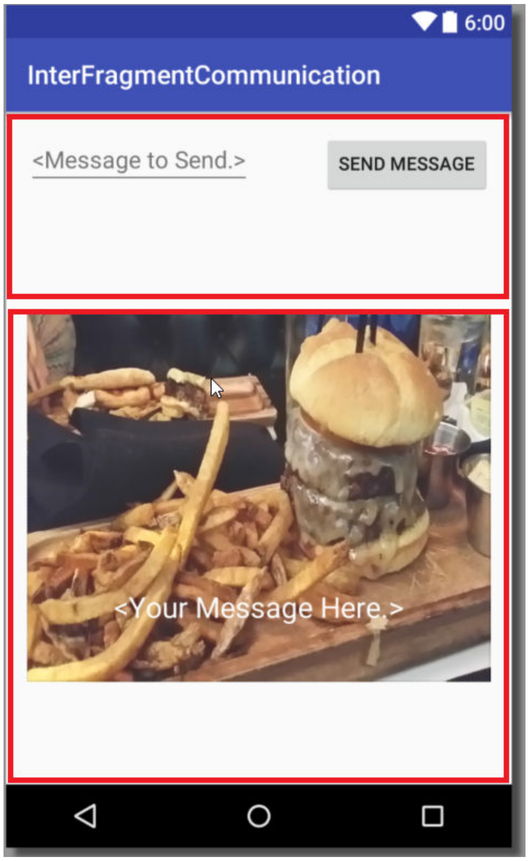
Team Members: ____

This worksheet is to be done in collaboration with your project team. Although we are working with Android, the concepts apply for any device.

Part I: Inter-Fragment Communication

1. **Implement an App with two fragments. The top fragment is the passer, the bottom is the receiver.** Whatever is passed from the top should be echoed in the TextView on the picture on the bottom. **REMEMBER:** you must pass requests between fragments through the parent activity, using an interface. This ensure "loose coupling". Fragments should not *talk* to each other.

Note: To save time, lock the orientation in portrait mode. Don't worry about saving state when the App is destroyed.



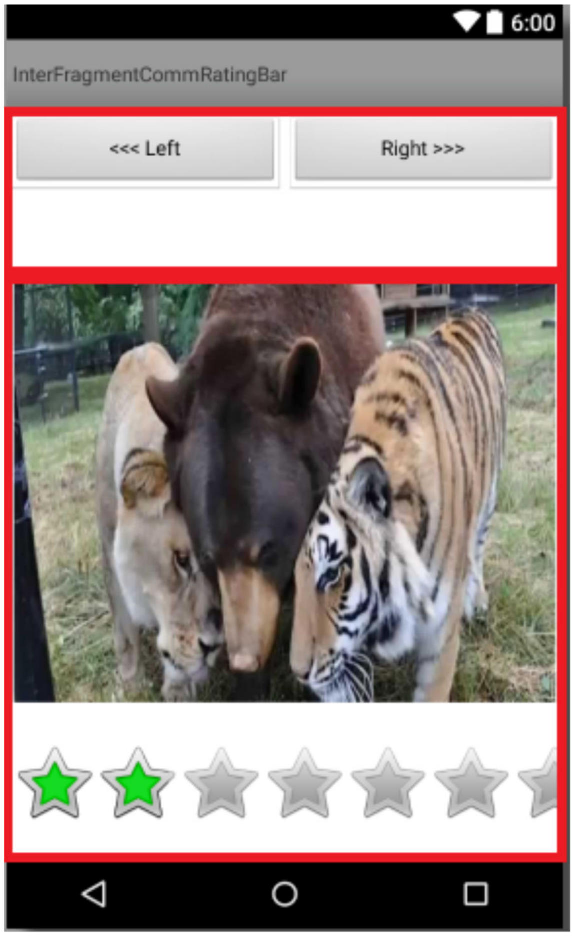
2. **Modify the above to use a simple ListView instead of the TextView. The list view will contain five items. When the user hits "SEND MESSAGE", the message as well as the image will get updated based on the description in the listview.**

   So for example if your ListView had the text "Hotdogs are Gross", replace the image to that of a Hotdog, and update the text as before. Get your own images for this exercise.

Part II: Inter-Fragment Communication Continued

1. **In lecture your instructor will give you code that will allow you to click Left/Right to change the image displayed (see below). The code uses only one fragment inflated into a single activity.**
   a. **Modify the code, separating the Left/Right buttons into their own fragment. Clicking left/right should work as before. The final Activity will have two fragments.**
   b. **Also add code to "remember" the number of stars for each image. That is, as you navigate between images the previously set number of stars should appear. Implement this any way you like, perhaps with an additional parallel array, or a new data structure you create yourself, containing the image and the star value. You choose!**

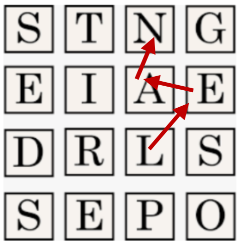
Note: To save time, lock the orientation in portrait mode. Don't worry about saving state when the App is destroyed.



Part III: Simple Boggle

Boggle is a simple word game that involves forming as many words as you can from a grid of letters. Sort of like a word search.

For example, given the following 4 x4 block of random letters, you can form the word "LEAN", by clicking on the letters, 'L', 'E', 'A', and 'N' in order.



S	T	N	G
E	I	A	E
D	R	L	S
S	E	P	O

A few other words in the example above: "STING", "PRIDE", "ANT", "SLANG", …

For this task you will implement a modified Boggle game,

Modified Rules:

- Your first letter can start anywhere.
- Subsequent letter touches must "touch" the previous letter in any direction.
- You may only use a letter in the current word once.
- You cannot generate the same word more than once, even if it's from different letters.
- Words must be at least 3 chars long.

Scoring:

- Each word is worth the same number of points as its number of characters, eg, a 3 letter word is worth 3 points.
- Incorrect responses reduce the score by 5 points, regardless of word length.
- There is no time limit.

Word List:

- Download a word dictionary to check answers.
- You can use any list of dictionary of words you like, many abound on the web, eg, https://raw.githubusercontent.com/dwyl/english-words/master/words.txt . No need to use a Database, just check if the users answer is in the dictionary.

GAME PLAY:

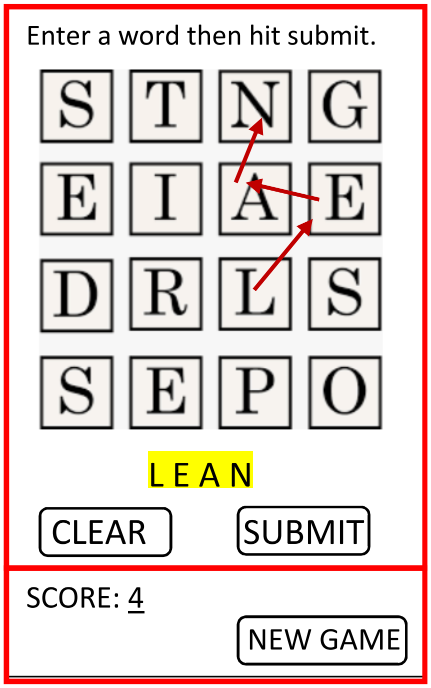
1. User selects a word in the grid.
2. User hits Clear or Submit.
3. If the user hits Clear, the word entry so far is removed with no penalty.
4. If the user hits Submit, the entry is checked against the dictionary.
5. If it is incorrect, 5 points are deducted from the score.
6. If it is correct, the proper number of points are added to the score, according to the rules above.

OTHER:

- Keep things simple, use toasts to communicate with the user, eg, “That’s correct, +4”, “That’s incorrect, -5”, “You may only select connected letters”, etc.
- Again, Users should only be allowed to click on contiguous letters.

IMPLEMENTATION REQUIREMENTS:

- Design the game using the template suggested.
- You must use fragments, these are identified with red boxes around the template.
- As discussed in class, fragments may not communicate directly.
- The two fragments must pass messages through an interface to the Main Activity.
- Do not create separate layouts for phones and tablets. **Just one layout in Portrait Mode.**
- Be considerate to the user. Apply simple “UI Polish” to the game, eg, after a letter is selected, color the button differently and disable it, so the user won’t accidentally select it again. You must think about other opportunities for making the UI user friendly yourself.



Enter a word then hit submit.

S	T	N	G
E	I	A	E
D	R	L	S
S	E	P	O

GAME TEMPLATE, PORTRAIT MODE ONLY