This lab will give you practice creating a multi-screen application. These are the main concepts you will apply:

* Starting a new activity using an Intent object
* Sending information from one activity to another using an Intent object
* Using activity life-cycle call-back methods: onCreate and onResume
* Using an ActionBar with “up” navigation to return to a “parent” activity

Part 1 – Tutorial: Hello Multiscreen Apps

Complete the tutorial, “Hello Multiscreen Apps”, which was also the required reading this week.

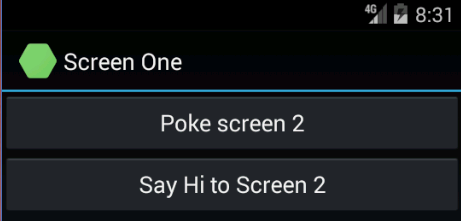
Part 2

**Assignment for Group A – Poke a Screen** (or poke someone or something of your choice)

Write an app with two activities. Buttons on the first activity’s UI will be used to send one of two kinds of data to the second activity. The second activity will display a message based on the data that was sent.

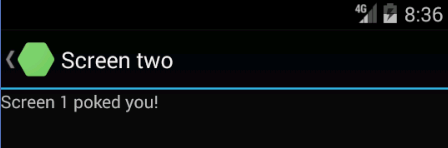
1. Add code to the default MainActivity

* Add a button with the text “Poke Screen 2”
  + In this button’s event handler, send an intent to screen 2 with a Boolean value set to true.
* Add a button with the text “Say Hi to Screen 2”
  + In this button’s event handler send an intent to screen 2 with the string “Hi from screen 1” (or some message of your own choice).



1. Add a second Activity

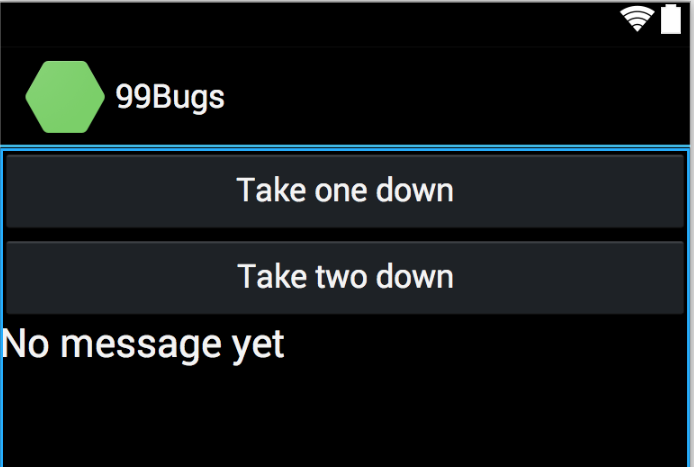
* Add an ActionBar with an “up” button.
* Add a TextView that displays messages from screen one. This one text view will either display the string sent in the intent or, if the intent contained a Boolean value, it will display “Screen 1 poked you”. (or a message of your choice).



**Assignment for Group B – 99 Bugs in the Code (or 99 other things of your choice)**

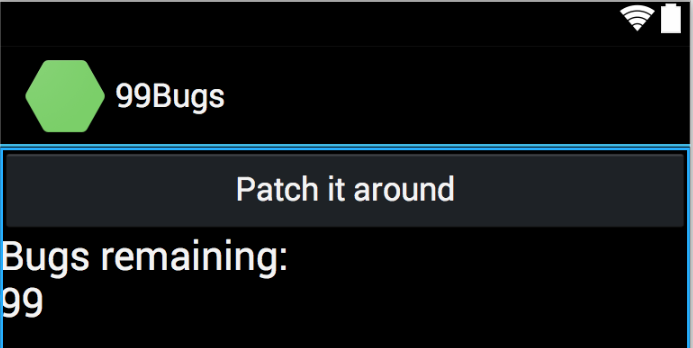
1. Add code to the default MainActivity (I’ll call this activity 1)

* Add a button
  + Change its button title to "Take one down".
  + Create an Intent that sends the number 1 to activity 2.
  + Add an appropriate event handler to the button.
* Add a second button
  + Change its button title to "Take two down".
  + Add an appropriate event handler to the button.
  + Create an Intent that sends the number 2 to activity 2. Start Activity 2 using StartActivityForResult
* Add a TextView and the code needed to display a message from activity 2. Get the message using onActivityResult.



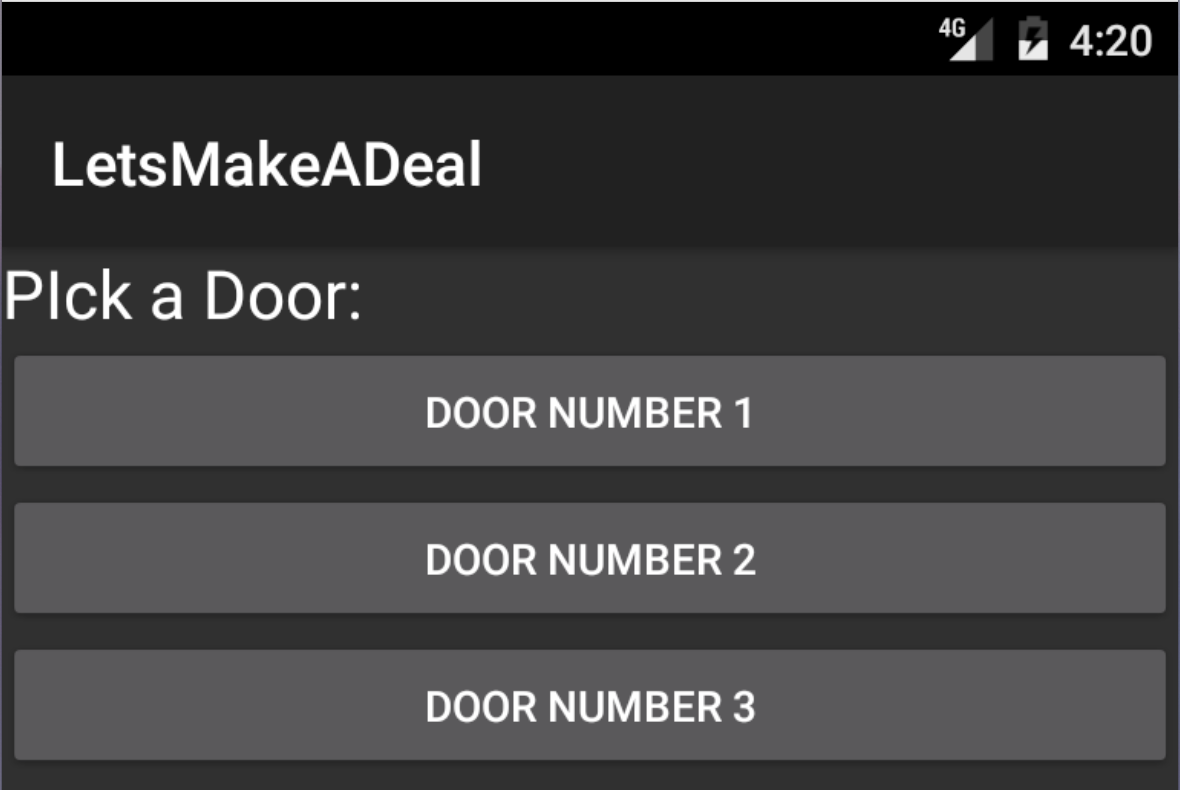
1. Add a second Activity and name it “The code”

* Add a button and a TextView to the second activity.
* The text on this button should be "Patch it around".
* Use an intent to send the number of bugs remaining to activity 1. It is important to send back the number not just to display it, but this is the only way to avoid losing the count of bugs.
* Add a button event handler and any other code necessary so that tapping a button subtracts one from the total number of bugs left, and sends the message above to the first activity.

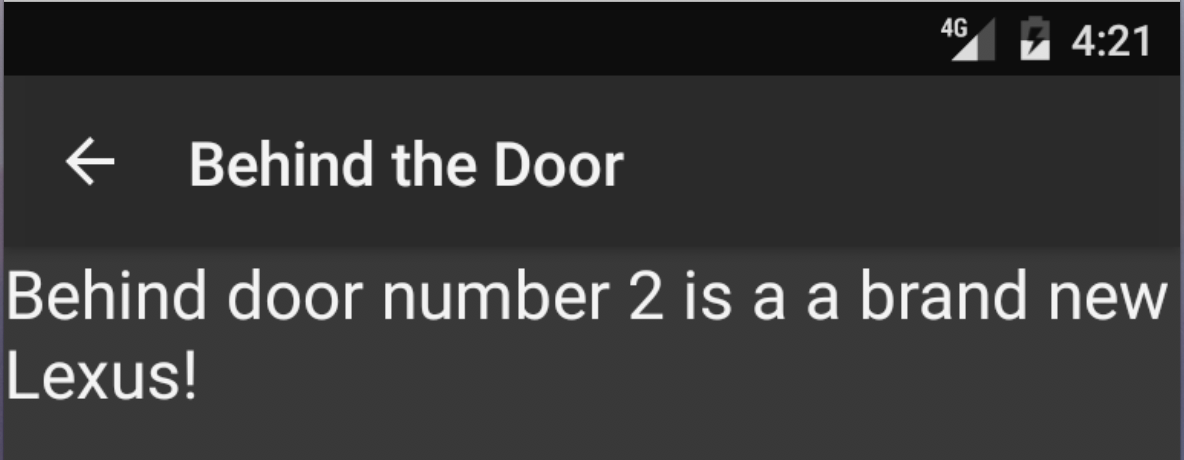


**Assignment for Group C – Let’s Make a Deal**

1. Add code to the default MainActivity

* Add a button with the text “Door Number 1”
  + In the event handler, send an intent to Activity2 with an int value set to 1.
* Add a button with the text “Door Number 2”
  + In the event handler, send an intent to Activity2 with an int value set to 2.
* Add a button with the text “Door Number 3”
  + In the event handler, send an intent to Activity2 with an int value set to 3.  
      
    

1. Add a second Activity

* Add an ActionBar with an “up” button.
* Add a TextView with the text: “Behind door number \_\_ is a:” using the number from the intent in place of t he blank
* Generate a random number between 1 and 3
  + Based on the random number append the text: “brand new Prius”, “goat”, or “skunk” in the TextView. (You can also make up your own prizes.)  
      
    

----------------------------------------------------------------------------------------------------------------------------------

**Submission to Moodle**

Beta Version

Post the following to the Beta + Code Review Forum:

1. For part 1: A document containing screen-shots of the tutorial app with each screen-shot labeled. (Please use .docx or .pdf format.)
2. For part 2: A zip file containing your app’s Visual Studio solution folder. (Make your solution smaller by deleting the *obj* and *bin* folders.)  
   Or, optionally, a link to a repository containing your solution source code. You can put the link on the same document with the report on your exercise from part 1.
3. A copy of your lab instructions (so the lab partner who reviews your work will know what your requirements were).

Production Version

1. Items 1 and 2 above, but revised as needed.
2. The code review of your work (the one done by your lab partner) with the second column (“Release”) completed by you.