Stanford University, CS 193A
Homework 1: Number Guessing Game, ... or not
(Individual; No Pairs)

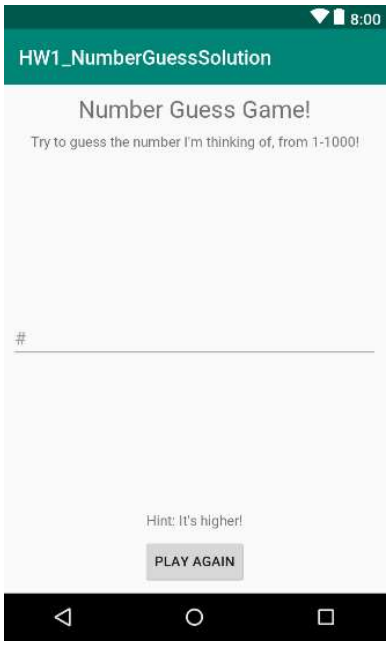
Assignment idea and spec by Marty Stepp.

The purpose of this assignment is to get a first taste of Android application programming and using the Android Studio editor, as well as using widgets/views and events to produce an interactive graphical application.

This is an **individual assignment**. Write your own solution and do not work in a pair/group on this program.

Program Description:

For this assignment, create a simple app where the computer thinks of a number from 1 to 1000 and the user tries to guess it. The app should give the user hints to help them find the correct answer. In particular, your app should have the following functionality:



- A way for the user to input numerical guesses.
- A hinting system so that after each incorrect guess, the app tells the user whether the right answer is higher or lower than their guess.
- Logic where the game ends (stops asking for / accepting guesses) when the player guesses the right number.
- At game's end, a display to the player that they won, along with a count of the number of guesses that were needed to find the right answer.
- A way to start the game over and play again once the game has ended.

Your app might look similar to the screenshot at right, though your appearance does not need to match this screenshot exactly. In particular, you are not required to use an `EditText` for the numeric input from the user if you prefer a different widget, nor are you required to use this exact layout and positioning of the widgets on the screen. You can make your app look however you like so long as it provides this basic functionality.

Name your project **cs193a_hw1_*sunetid***. Name your activity **NumberGuessActivity** and your layout file **activity_number_guess.xml**.

Don't Want to Write a Number Guessing Game?

Okay, fine! Since this is the first assignment, we are willing to be super flexible. If you don't want to write this program, you can make **any Android app you want** and submit it to receive credit, so long as it meets the requirements outlined below. Below we list some suggestions of app ideas. You can pick any one of these, or you can come up with your own creative idea and submit it, so long as it meets our constraints. (Some of the ideas below are trickier than others or use material we haven't covered yet, so you'd have to do your own digging.) Ask us if you're unsure.

- **Rock-Paper-Scissors:** The human and computer players each pick an option of Rock, Paper, or Scissors. The computer's choice is made randomly. Paper beats Rock; Scissors beat Paper; and Rock beats Scissors. Keep track of the human player's score against the computer over time.
- **Grade computer:** Type in your homework and exam scores and it estimates your grade in a class.
- **Tip calculator:** User types in how much money they spent on their meal at a restaurant and chooses a percent to tip, and the app outputs how much money to leave as the tip.
- **Memory game:** (hard) Several buttons are shown on screen, "face down", and the user needs to try to find pairs that match up when flipped over.
- **Hangman:** (hard) Computer thinks of a word and the user has to guess letters to try to reveal it.

Functionality Requirements:

Regardless of what type of app you choose to build for Homework 1, we have a few necessary requirements that you must follow. These are important to us, both for making sure you practice the right skills, and for helping facilitate the process of grading. If you do not follow all of these requirements, you will not receive full credit for the assignment.

- Your app must be set up as an **Android Studio project**, so it can easily be opened/run/graded by others.
- Your project's name must be exactly the following, case-sensitive, including the underscores:
  - `cs193a_hw1_SUNETID`

  where ***SUNETID*** is the part of your Stanford email address before the `@stanford.edu`. For example, if Kelly Smith with email address ksmith12@stanford.edu submits her Homework 1, her project name MUST be exactly the following:

    - `cs193a_hw1_ksmith12`
- Your app must use at least **3 widgets**/views on the screen. (Example: Two `Button`s and a `TextView`, ...)
- Your app must use at least **2 different kinds of widgets**. (Example: `Button`, `TextView`, `EditText`, ...)
- Your app must respond to at least two different **events**. (Example: clicks on two different buttons.)
- You must change at least some aspect of the physical **styling** or appearance of some widgets in your layout. For example, make a text view have a larger or bold font, or make text appear in a blue color.
- We haven't talked very much about **layout** yet, so your app can have essentially any layout you want, so long as the various widgets are visible and can be interacted with on a standard AVD emulator device for a phone.
- Your submission must be **your own work** and not written by someone else.
- Your submission must be **new work** for CS 193A and cannot be a project you submitted for another class, such as CS 108 or CS 147.

Style Requirements:

This course will generally have more lenient coding style requirements than a class like CS 106A/B. But we do have some important style requirements below that we ask you to follow. If you do not follow all of these requirements, you will not receive full credit for the assignment.

- **Comments:** Write a **comment header** in your main activity .kt file containing your name and email address along with the name of your app and a very brief description of your program, along with any special instructions that the user might need to know in order to use it properly (if there are any). Also write a brief comment header at the top of every **function** in your .kt code that explains the function's purpose. All of these comments can be brief (1-2 lines or sentences); just explain what the method does and/or when/why it is called. You do not need to use any specific comment format, nor do you need to document exactly what each parameter or return value does. The following is a reasonable example of a comment header at the top of a .kt activity file:

```
1 // Kelly Smith <ksmith12@stanford.edu>
2 // CS 193A, Winter 2049 (instructor: Mrs. Krabappel)
3 // Homework Assignment 1
4 // NumberGame 2.05 - This app shows two numbers on the screen and asks
5 // the user to pick the larger number.  Perfect for Berkeley students!
6 // Note: Runs best on big Android tablets because of 1000dp font choice.
```

- **Redundancy:** If you perform a significant operation that is exactly or almost exactly the same in multiple places in your code, avoid this redundancy, such as by moving that operation into a helping method. See the lecture code from our lecture #1-2 for examples of this.

- **Reduce unnecessary use of global variables and private fields:** While you are allowed to have private fields ("instance variables") in your program, you should try to minimize fields unless the value of that field is used in several places in the code and is important throughout the lifetime of the app. Our lecture code from Week 1 shows examples of reducing fields by accessing values from widgets instead (using `findViewById` and `text` and similar).
- **Naming:** Give descriptive names to variables and functions. For example, other than a `for` loop counter variable such as `int i`, do not give a variable a one-letter name. Similarly, if you give an `id` property value to a widget such as a `TextView`, apply a similar style standard as if it were a variable name and choose a descriptive name.

Outside of the constraints above, you can do most anything you want! Again, we're trying to let you be creative here, and we won't be quite as strict as CS 106 is about grading or style in general. Your app does not need to be totally unique; if you have an existing idea that you want to replicate, or if you and a friend want to make the same kind of app, please go ahead.

If you don't do the number guessing game, we recommend you do something simple for this first program since you will need to set up the Android Studio IDE and emulator, and you'll have a lot of new syntax and features to get used to. These assignments, as well as this class in general, are meant to be **low-stress** and fun. If you want help, please feel free to ask for help in our online **Piazza** forum and/or come to office hours. Feel free to make an app as simple or as complex as you like, relative to your familiarity level and time constraints.

Turning In:

You must pack your homework project into a ZIP archive and submit it using our "Grade-It" grading system. See the following page for general instructions about how to ZIP your project properly and how to submit to our system:

- Homework turnin instructions (general)

You can find a direct link to the "Grade-It" turnin page for this assignment back on the Homework page of the class web site.

Survey: After you turn in the assignment, we would love for you to fill out our optional anonymous CS 193A homework survey to tell us how much you liked / disliked the assignment, how challenging you found it, how long it took you, etc. This information helps us improve future assignments.

Honor Code Reminder: Please remember to follow the **Honor Code** when working on this assignment. Submit your own work and do not look at others' solutions. Also please do not give

out your solution and do not place a solution to this assignment on a public web site or forum. If you need help, please seek out our available resources to help you.