Project 1 - Simon

You will create a “Simon” application that will run on an android emulator/device. This app will have 4 colors on the screen that will randomly light up in which the user must repeat. Every round should increase the number of lighted up colors but retain the previous sequence and adding a new “random” selection to it. This will continue until the user presses the incorrect selection.

Wiki link : https://en.wikipedia.org/wiki/Simon_(game)

Requirements

1. This app will have a total of 3 screens: Start, in-game, Results
2. The game will contain 3 difficulty levels.
    1. This will be the “**Start**” screen.
    2. This screen will be the entry point into the application (your “main”)
    3. What the difficulty will determine:
        1. Shorten the amount of time they have to select
        2. Increase the speed of the “button” flashes
        3. Start with an initial increase in the sequence (easy: 1, medium: 3, hard: 5 for example)
3. Once selected, move to the next screen.
4. When a user selects a color, provide feedback (Depressed color change or something).
    1. This is your “**in-game**” screen.
5. Your screen must contain 4 distinct, **selectable** colors.
6. If the user fails to follow the correct sequence, the game is over.
    1. This is the “**Results**” Screen
7. The user must be presented with:
    1. A “**Game over**” message to your liking,
    2. Their score for that round, and
    3. The ability to start a new game (this could, for instance, finish() all activities and take them to the “**Start**” screen again).
        1. Need to clean up your resources. (Activities, Model Data, etc.)
8. We will NOT focus on orientation changes. Use your manifest to lock the orientation into portrait or landscape only.
9. SEPARATE YOUR CODE! (this is very important)
    1. Follow MVC design patterns as much as possible

2. Do not store data in a View/View Fragment or in an Activity (this means no variables that hold data)
3. All data must be in a Model object of some form

How the game will work:

1. You will retain a dataset of the previously generated colors and compare the corresponding selection to what should be in the generated list. Example:
    1. System randomly picks **green** and stores it. User selects **green**.
    2. System randomly picks **red** and stores it. User needs to select **green**, then **red**.
    3. System randomly picks **red** and stores it. User needs to select **green, red, red**.
        1. At this point your stored list should be green, red, red.
2. If the user selects the incorrect color there should be feedback of the color that was suppose to be pressed (highlight the button view somehow) and then be presented with a “Game Over”.
    1. You will have to figure out how you can delay the presenting of the “Game Over” screen so that the user sees the color that was supposed to be pressed.
3. User should have a high score on the screen while playing the game.
    1. Persist this through the session of the app at minimum (This part is required).
    2. **Bonus**: Find a way to store the data where you can shutdown the app and restart and maintain a history of scores (say, 10 or so). In order to get this, you will need to also create a new screen that would display this data.

You will need to comment your code. I need to know that YOU know what you are doing and that you are the one that did it. This project is worth 100 points and is due on **Wednesday February 28th**. This is about 3 1/2 weeks so it should be plenty but don’t start in the 3rd week!!!! (you will wish you didn’t) You may use any resources you have at your disposal but no direct copying and anything that is obtained from someone else must be sourced (I don’t need a source like you put in a report, I need a // I obtained from xyz.com/filename/etc within the source code file it is used.)