Justin Victoria

Professor Sheng

Android

20 December 2019

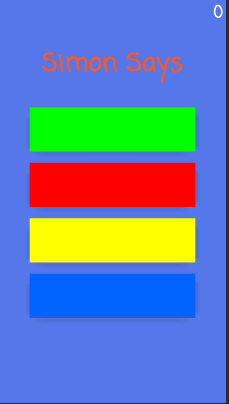
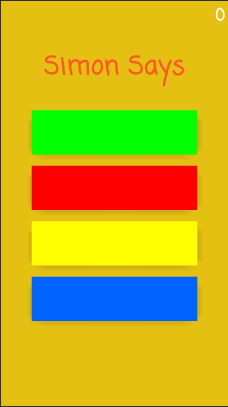
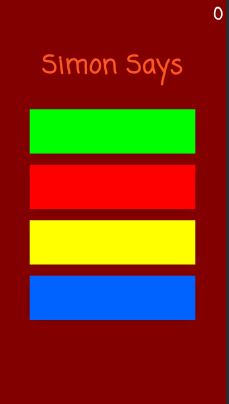
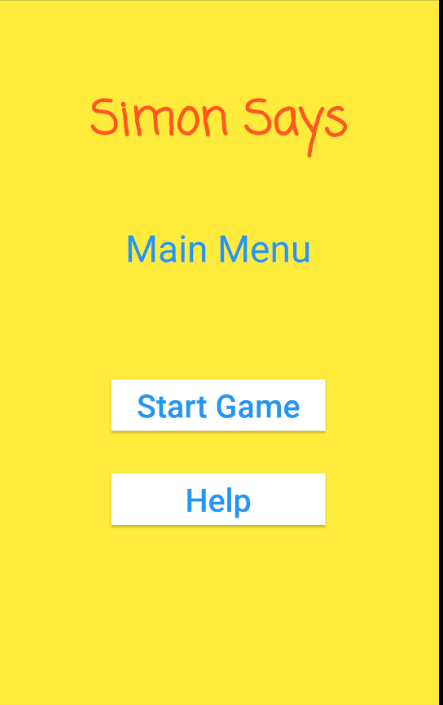
Project Description Document

**1. Project Statement**

The app I wanted to develop was an app that would act as a version of the game “Simon Says”, which is a memory game where the player has to follow the action patterns of the app and if the user fails to do so, loses. The app has a menu with a title, start button, and help button. The start button would bring the user to game and the user would see buttons of 4 different colors shown. The user would have to tap the button corresponding to the color of the background, but if the prompt at the top says “Simon Says” followed by a color, the the user has to tap the button that corresponds to the color described in the prompt. If the user fails to do so, the user loses. I decided to develop this app for the sake of entertainment and because I thought it would be fun to develop it. The kind of users who would use this app are those looking for a little fun game to play. Such app already does exist on the app store, and there is no special requirement for using the app other than the ability to see and tap on a screen.

**2. Application Design**

Title Screen Green, Red, Yellow, and Blue Screens



MainActivity.java

Green.java

Yellow.java

Blue.java

Red.java

There are a total of 5 activities and the MainActivity acts as the common activity the other color activities link to through the use of intents. It will go to a random color activity when the start button is clicked. Within each color activity, there are intents that will link to another color activity if the game isn’t over yet. The .xml files were implemented in a fashion that would connect the components accordingly.

**3. Application Implementation and Evaluation**

The app was of course, implemented in Java. As mentioned earlier, I have a MainActivity class, along with a Green, Red, Yellow, and Blue class to correspond to respective color screens for the game. Within the color classes are also a few helper functions and all classes have listeners in use. When it came to testing the app, I tested it by running it and seeing how it would be to use the app. I also tested using the debugger. The app initially did not work quite correctly since the app stopped working after clicking on the start button. I suspected that the reason may have had to do with the part of the code that updates the score in each color class. I found that to be the case. Another issue I had to fix was a mislabeling in my .xml files. I later found that I needed to fix my initialization of the private activities array in the color classes and found that my count variable was assigned to -1 rather than decrementing by -1. After addressing these issues, the game worked as desired.

**4. References**

The Android Studio Developers guide and YouTube.

**5. Experiences and Thoughts**

Unfortunately for me this semester, I had a tough balance of coursework along with a wonderful internship at a company, so in general I was very time crunched. I didn’t get around to taking the time to implement the part after clicking the “Help” button since I needed to focus on the code that comes after clicking the “Start Game” button. The class can be improved by either setting designated quiz dates so people know when they’ll be, or if not, when it comes time to notify students, do so by sending out an email rather than simply posting an announcement on Blackboard very close to the date of the quiz, since most people don’t check Blackboard announcements right away.