Pistachio

Deliverable: Iteration Two

Team: Pistachio

Members: Cameron Dziurgot, Luke Brodowski, Travis Moretz

Submission Date: 19th November 2015

1. Iteration Functionality:

After the second iteration the application visually, has the same visuals as the first iteration. The application now stores sequences entered by a user in a folder that can be pulled off of an emulator of device and tested in various comparison algorithms.

The representation of tap has changed in our business logic and now all timestamps are relative to an absolute time of zero. This was something that wasn't thought of during the first iteration but the change had to be made to make the averaging and comparing of the sequences possible.

Our averaging algorithm takes three tap sequences, averages them and obtains a standard deviation between the three sequences. Once an average master sequence is created, the comparison algorithm can be used with a experimental sequence along with the standard deviation. The algorithm will return a dissimilarity score which indicates how close the master sequence is to the experimental sequence. The closer the score is to 0, the greater the similarity between the two sequences.

2. User Stories Implemented:

1. Setting a passkey: estimated complexity 3.
    a. creating averaged passkey attempt: 2
    b. saving passkey: 1
2. Testing a passkey: estimated complexity 3.
    a. getting attempt from user: 1
    b. comparing to saved passkey: 2

3. Iteration Changes:

For this iteration, a user will still be able to create a passkey sequence by entering multiple sequences. During this process, the sequences are tested to make sure they have the same number of taps. Once this is done, an average between the sequences is created and saved with the trial sequences.

Once the master sequence has been created, it can be compared to an experimental sequence a user enters. This comparison yields a score which allows us to identify if the master sequence and experimental sequence are close to some degree of confidence that we have yet to decide. Depending on the degree of confidence and the score, we can decide with confidence whether to grant the user access or to deny the user access.

4. Lessons Learned During Iteration:

Business requirements change and it's important to be adaptive and be able to rework your code to reflect those changes. For example the way we were storing taps had to be changed because we needed certain data that wasn't being recorded at all. We also had to change the timestamps to be relative to an absolute time (the first tap) that way the change in system time doesn't have to be taken into account when averaging, or comparing a sequence(s). This makes the tap sequences system independent. This refactoring made it easier to compute averages, and easier to compare two different sequences.

5. Updated list of User Stories still to be Implemented:

1. Viewing Statistics: estimated complexity 5
    a. creating statistics algorithms: 2
    b. adding new attempt to averaged statistic: 2
    c. displaying current statistics: 1
2. Escaping Statistics: estimated complexity 1
3. Routing all views together: estimated complexity 3

6. Next iteration Implementation:

For our next iteration and implementation everything will be brought together into a single application. The biggest change will be the ability to compute statistics on passes/fails of user entries. This will involve creating the actual algorithm to compute the statistics, a way to save and record these statistics and finally a way to display the current statistic.

Another thing that needs to be done to route all the views/main activities together so the entire app functions. That way the user will be able to register a tap sequence, go back, try to login, tap the experimental sequence and then view system the statistics on pass/fail attempts. This will encapsulate our previous 2nd iteration where all the algorithms for averaging and comparing will be inserted into the actual android application.