Mobile Perimeter

PREPARED FOR:

ECE 1778

University of Toronto – department of computer and Electrical engineering

Specialist: Professor Moshe Eizenman

Developers: Irina Kon – 994757786
Joseph Azzam - 1001972626

Date: April 3, 2016

Word Count: 2292

Table of Contents

Introduction

Visual Field Testing (Perimetry testing) is done to measure a patient's sensitivity to light and enables detection of visual impairment, specifically the detection of Glaucoma. During the test, flashing stimuli with different intensities are projected on a darkened screen and the patient indicates when they are able to perceive the light.

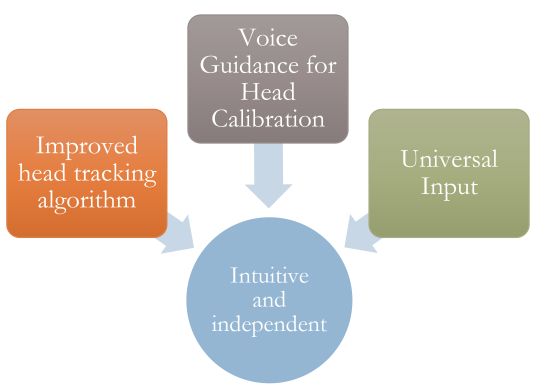
The goal of the Mobile Perimeter project is to develop a visual field test on a mobile device. The value of such a test is to

a) Enable patients to perform these tests at-home independently and
b) To help provide developing nations with an affordable visual field testing instrument

The Mobile Perimeter is not meant to replace the existing testing machine, called the Humphrey Field Analyzer (HFA), but rather help fill the gap due to the expense and immobility of the HFA.

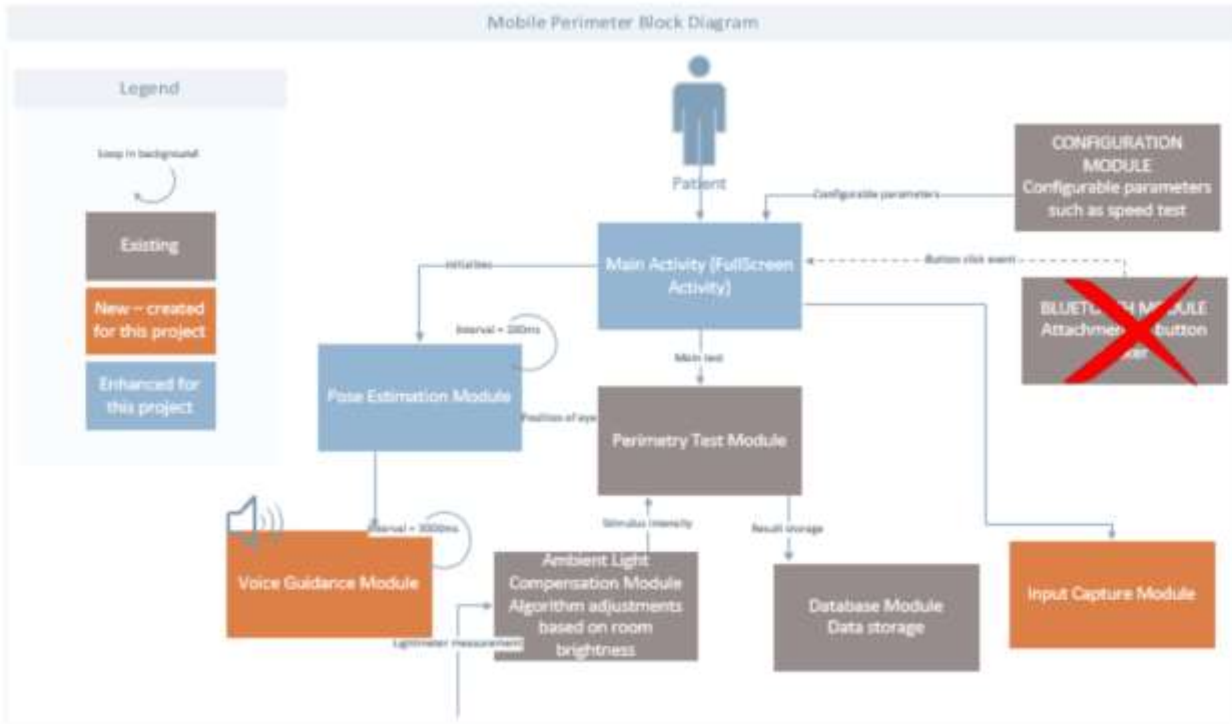
Building upon an existing proof of concept (built by Ilona Wong for her Master's thesis), our group worked to enhance the prototype with a goal of making it feasible for the app to be used independently by making the application more intuitive. Our enhancements were based on the limitations of the mobile perimeter as outlined in Ilona's thesis and focus on three main objectives:

a) Voice guidance to help guide the user into position thereby improving the user experience as the original wording on screen was difficult to see and follow
b) Enhancements of the head tracking algorithm to facilitate correct voice guidance
c) Universal input capture to remove reliance on proprietary clicker (SensorTag) and allow usage of any universal Bluetooth clicker.



Overall Design

Presented below is our block diagram. Modules outlined in grey are existing modules which have



Main Activity (FullScreen Activity) – Enhanced

This is the main activity of the app. This activity initiates the camera frame capture loop and initializes the other views including the menu view, positioning view and perimetry test views.

The User Interface has been redesigned with our main goals in mind: ***intuitive*** and ***independent***. The UI was updated with larger fonts as well as higher contrast that will maximize the probability that visually impaired individuals are able to complete the test independently. The pattern has been added to allow patients to download the pattern directly from the app.

Configuration Module

Provides the configurable parameters (including hardcoded hardware parameters) for the user to set including the test of the speed, pattern size, head tracking functionality etc. Configuration for toggling positioning text on/off has been added.

Perimetry Test Module

This module contains the test algorithm while taking into account the post estimation as well as the ambient light compensation. It also keeps track of the result of the test.

Ambient Light Compensation

This module takes into account the eye position from the pose estimation module as well as results from the camera's light meter to determine the intensity for the displayed stimulus.

Database Module

This modules stores the patient's result in the database.

Bluetooth Module

We have removed this module from the app and replaced it with the Input Capture module to resolve an existing pain point in the app of the user having to use the SensorTag proprietary clicker.

Input Capture Module

Our improvements to the app included replacing the Bluetooth module with two options: a voice command recognition system as well as the ability to use a selfie-clicker button or headphone volume up button as a backup. Selfie clickers and headphones are commonly available and are more intuitive than the SensorTag which was used previously. Voice recognition was put in as the best alternative as it requires no additional peripherals. The voice recognition module uses the Pocketsphinx library and uses speech patterns to recognize the user's input. Although the best alternative in terms of usability, the voice module's performance was not conducive for use in the app. User input was registering too slow and required the use of sentences for 100% correctness which hindered the smoothness of the app. Additionally, it took extra load on the tablet which created stutter performance.

Pose Estimation Module

Head tracking is done by identifying the location of an asymmetrical circles pattern which is positioned on the patient's glasses over the eye that is not being tested. This module obtains the camera frame and through 3rd library processing (OpenCV) determines if the pattern was found. The existing algorithm for tracking used a simple counter to keep track of number of times pattern was not found and forced the user to re-calibrate after the threshold was reached. While troubleshooting the app, it was discovered that the reliability of finding the pattern is quite low and therefore the guidance presented to the patient was frequently outdated and not accurate enough. We enhanced this module to use an exponential decay algorithm so that more recent pattern matches have higher weight but previous ones are also taken into account. Instead of using the last result of when the pattern was found, the algorithm now decays exponentially and based on the number of failures versus successes determines whether to indicate to the user that the pattern wasn't found.

Voice Guidance Module

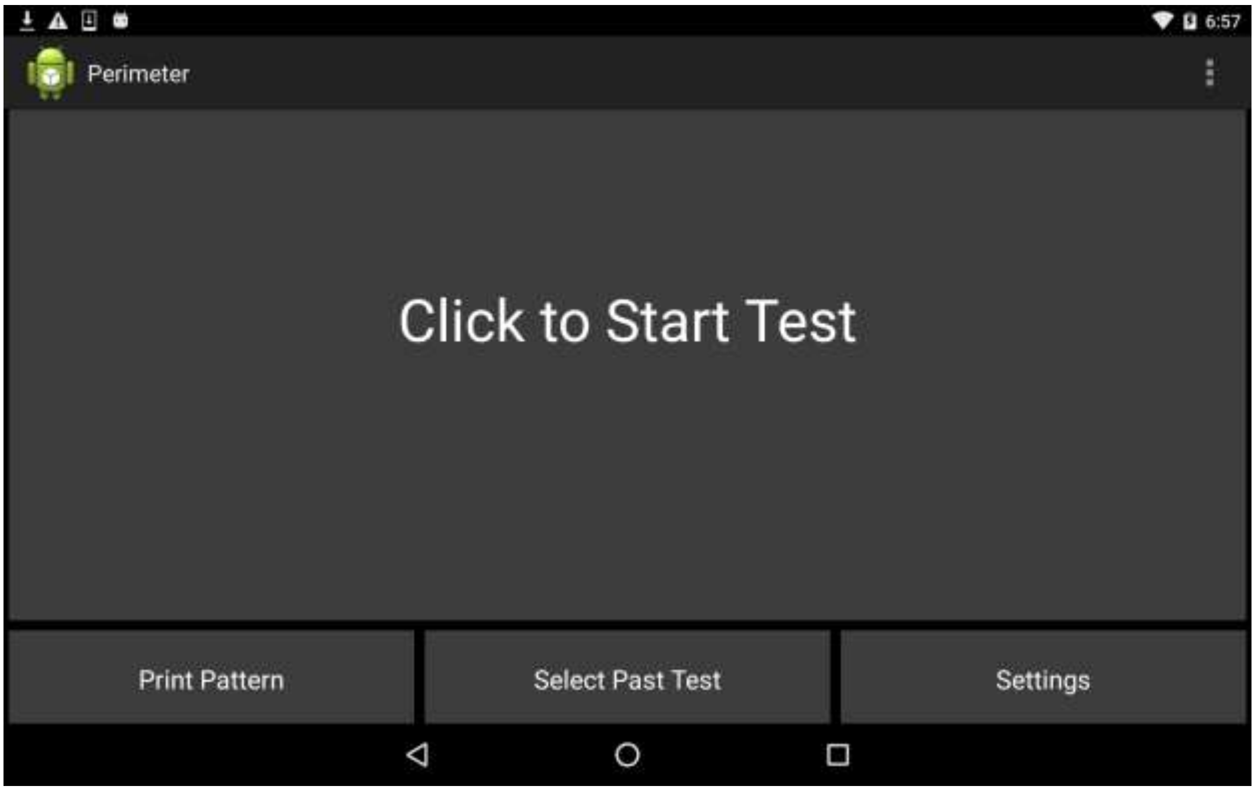
This module provides voice guided feedback response to the user for positioning purposes. Voice commands tell the user how to adjust their position to ensure they are within the range of the camera. This module was introduced by us into this project to build on the issues with the on-screen text guidance instructions. The on-screen text was an issue for patients with visual impairment and wasn't intuitive. While the existing on-screen drawing module could redraw the instructions every 300 milliseconds, voice tracking has a unique challenge where the frequency interval cannot be less than 3 seconds. Speaking at the user more frequently produces stuttering results. This challenge forced us to evaluate a better accuracy algorithm for the actual pose estimation as described above.

The voice guidance module was also implemented to run in the background during the perimetry test. In the existing application, only pattern identification was available but the user's head was not being tracked. By adding this functionality behind the scenes while the test is being taken, we significantly reduce the risk of the patient moving out of field of view and having to recalibrate during the test (which can be startling to the patient). If the patient slowly starts moving out of view while taking the test, voice guidance will guide them back into position.

Statement of Functionality

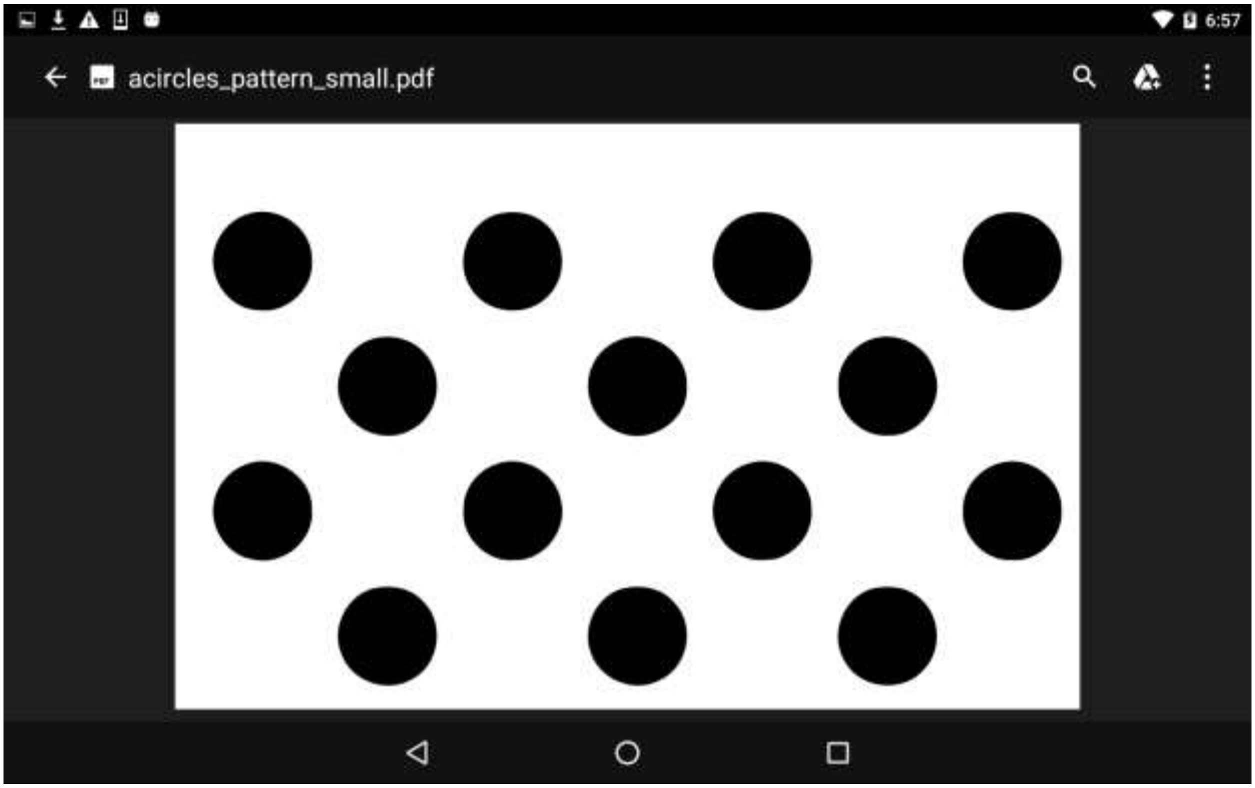
Landing Screen

The landing screen provides a clean interface for the user to take the test. The start button occupies the majority of the screen as it is expected that the user will mostly be using this functionality after initial setup. The interface is high contrast being mindful of the user base having visual impairment.



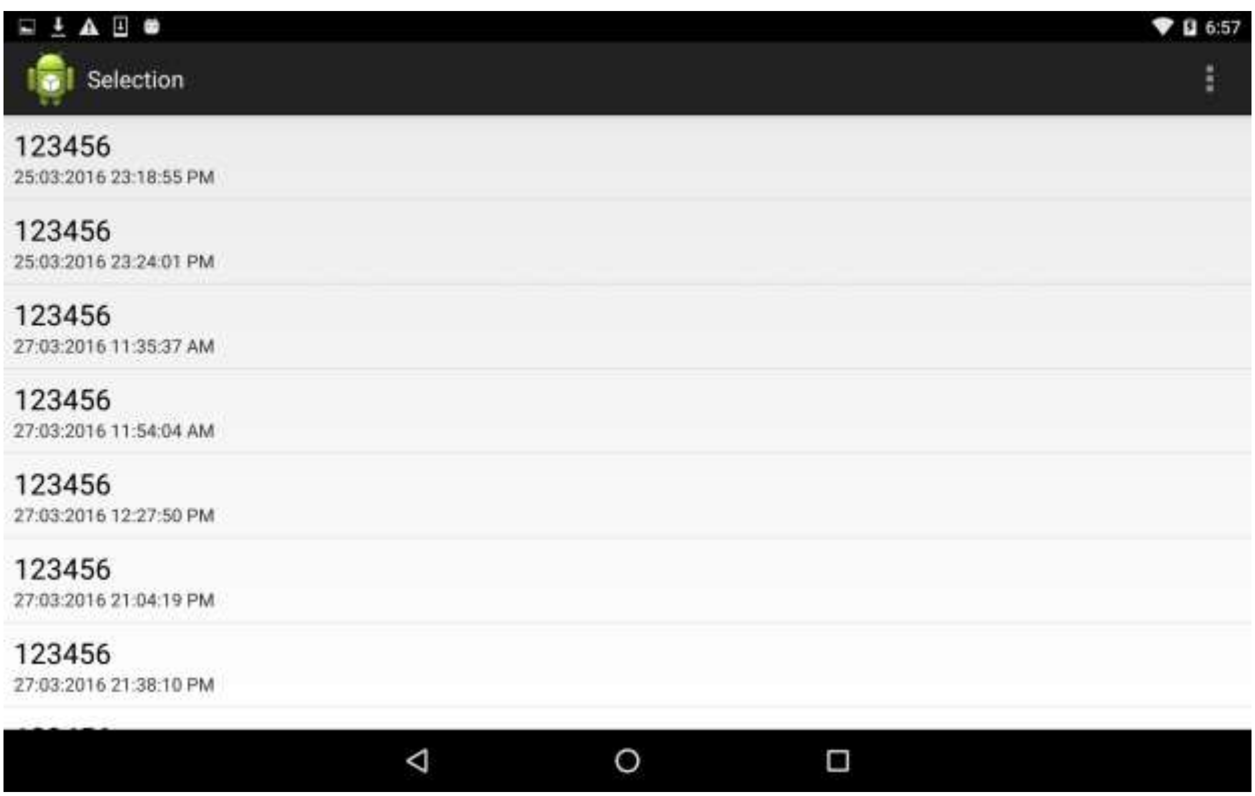
Print Pattern Screen

The print pattern functionality allows the user to print the pattern directly from within the app. This removes the reliance on an external source to provide the pattern and makes the app more self-sufficient.



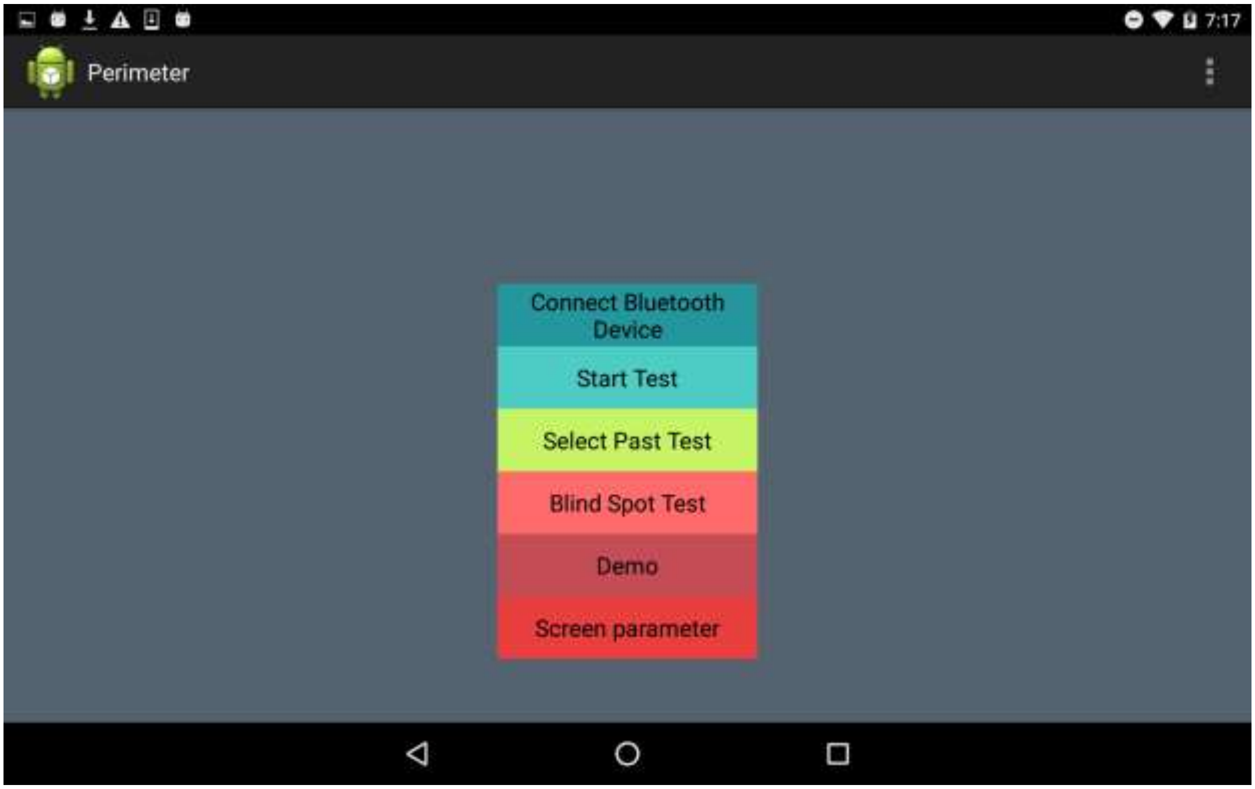
Select Past Result Screen

This screen provides the user the ability to view past results.



Settings Screen

This screen navigates to the existing application's landing page.



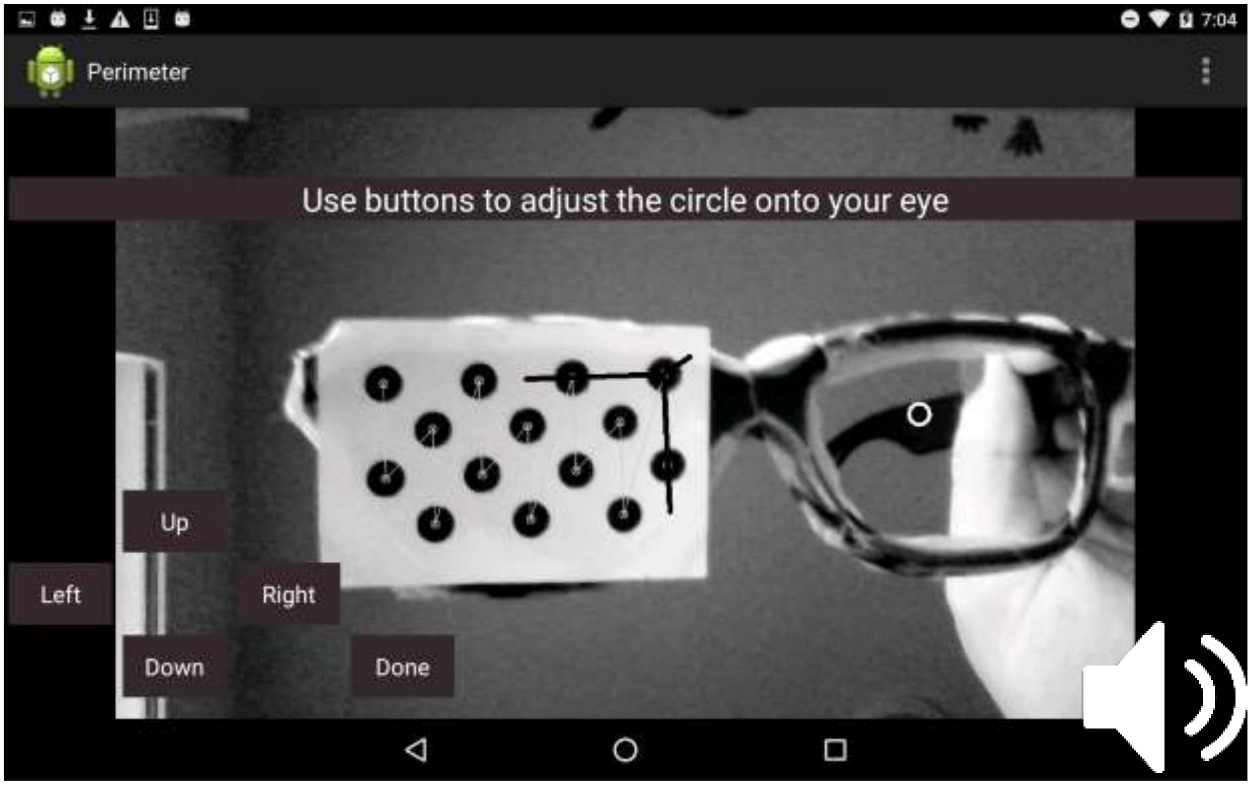
Start Test/Initial Positioning Screen

This screen is presented to the patient once they click on start test. Voice guidance provides instructions for the patient to position the circle over the eye to be tested. The patient performs the task and clicks done.

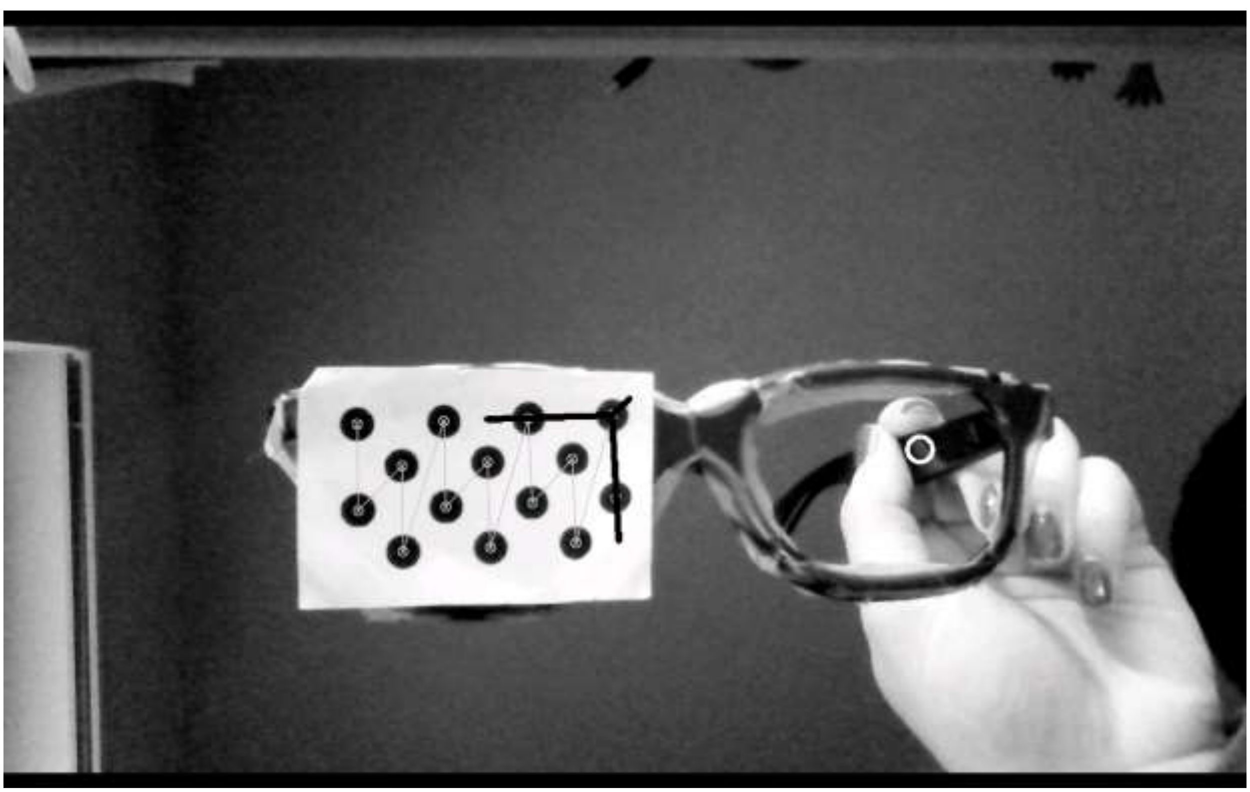
This screen has also been redone to provide a high contrast interface.

Start Test/Initial Positioning Screen

This screen is presented to the patient once they click on start test. Voice guidance provides the patient instructions to position the circle over the eye to be tested.

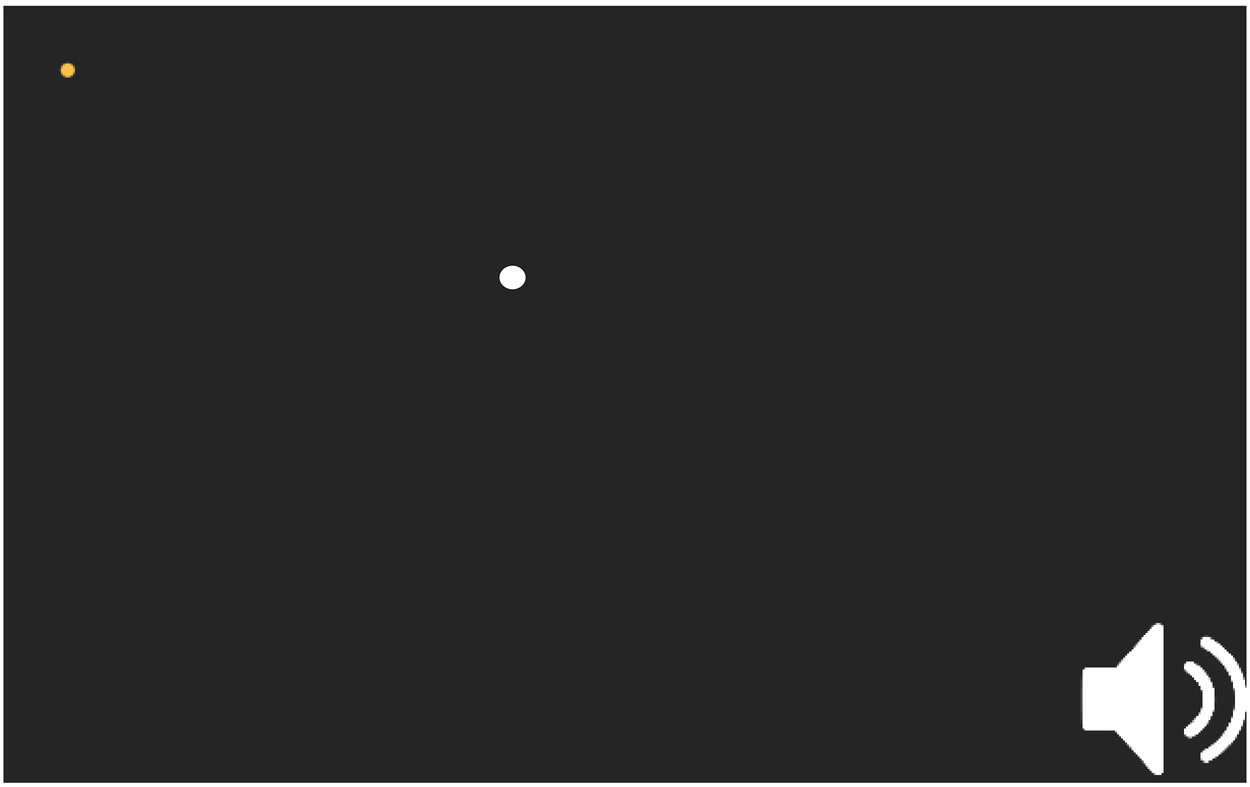


Once the patient clicks done they are presented with the initial positioning screen. The application voice guidance guides the user into position by specifying verbal instructions to move closer, further, higher, lower, left or right from the screen.



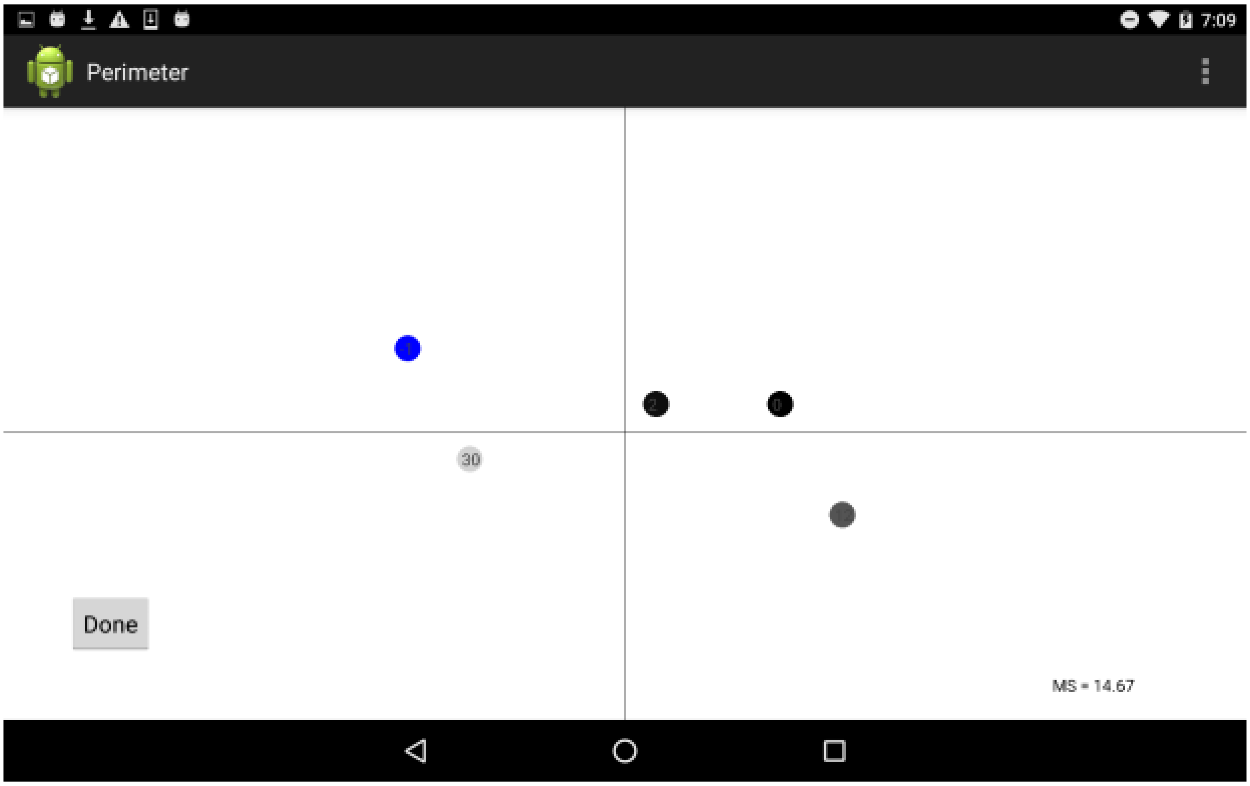
Perimetry Test Screen

This is the main exam screen of the application. It shows the user stimulus on screen and registers the click based on the users input capture method. Upon receiving input, the application provides the patient with sound feedback in the form of a beep. This voice guidance module works in the background to keep track of the user's position and notifies the user to get back into position if they start drifting away from the optimal positioning for the accuracy of the test.



Test Result Screen

Upon completion of the test, the user is notified the test is complete via voice guidance and the results are presented via a graph.



Lessons Learned

We learned many skills including both soft and technical. From a soft skills perspective, we learned how to gather requirements from a specialist as well as the ability to work in a fast-paced project environment. We obtained some domain knowledge in Ophthalmology and we learned how to set small goals as well as work towards a larger common goal.

From a technical perspective, we learned how to enhance existing software. We learned how to troubleshoot complicated concurrency issues and when to ask for external help to ensure we don't get stuck on any one item for too long. We learned to read and work within an existing code base and ensure the changes we make don't break existing software. We learned to fix issues as they so frequently arose. Learning how to use external libraries and native C++ code and JNI to call OpenCV functions, we also got more familiar with the Android Camera APIs, the Bluetooth functionality, voice recognition and signal processing.

We also learned that in some instances, just because a solution seems good in theory does not mean that it has the correct metrics needed for the app to function properly and correctly. For example, we spent effort developing the voice command system to register user inputs via a single syllable word (e.g. yes). This is a great idea in theory as it would eliminate the reliance on an external input mechanism but in practice and upon testing proved to not be efficient, fast or reliable enough. As such, we learned that even when something makes sense from a technical stand point, it may not always meet the non-functional requirements of the app.

Another key learning was time management as the deadlines for this course were quite aggressive.

Member Contributions

Irina

- Setup the subversion repository for the project and performed the necessary fixes when builds got broken
- Migrated the application to Android studio from the eclipse project while also upgrading OpenCV to the latest version to take advantage of performance enhancements (this involved resolving many issues and fixing dependencies).
- Worked with Ilona to understand how to get the app running beginning to end and troubleshoot all issues as they arose while preparing the demos for the in-class presentations
- Added the voice guidance functionality
- Rewrote the head-tracking algorithm to use exponential decay for more improved accuracy while working to resolve concurrency issues with the existing app
- Added feedback mechanism that indicated to the user that their response has been registered
- Redesigned the UI to make it more user friendly for individuals with visual impairment

Joseph

- Assigned the task of finding a replacement for the Bluetooth clicker
- Contributed research by investigating viable speech detection alternatives and their limitations
- Created a voice command system for the user interface, in addition to voice trigger for the test, and introduced the selfie clicker as a replacement
- Worked on optimizing the voice recognition for better accuracy of single syllable words which were needed for the specific application of this app
- Created a standalone loud detection app and evaluated the viability of this solution within the constraints of the app
- Researched the feasibility of using google cardboard as a potential alternative and put together a sample app

Future Work

The current app, with our enhancements, is still very much a proof of concept. From a camera perspective, the app could be enhanced to be faster so that the user feedback is continuous and not choppy. It will be crucial to port this application to a larger device. This will allow the patient to not have to flip the device during the test and better logic could be used to increase the efficiency of the test. Additionally, with so many concurrent runnable background tasks, optimization of all modules will be vital to ensure smooth performance.

From an algorithm perspective, through some basic testing we discovered the exponential decay algorithm to be providing decent accuracy but thresholds and weights could be improved to obtain even better accuracy.

For speech recognition it is evident that the detection speed currently is not enough for smooth and accurate results. Additional may need to be performed to optimize the detection speed and frequency.

Some better solutions can also be explored that will eliminate the need for positioning all together. One major one of benefit may be Google Cardboard as presented in our findings. It is affordable, will allow for an enhanced screen size and a fixed head position that may eliminate the need for tracking.