Personal Report – Group Project

Team Hotel – Eye-Tests on Demand

Jan Ondras
Trinity College
Part 1B
2015/2016

My contribution

My main task was to learn about Staircase testing procedure and implement it in our eye-testing method. I was part of the server-side team together with Gabor.

- **Design**
  During the design phase I contributed mainly to Project plan and server architecture: annotated interfaces (see Requirements & Specification). Then I concentrated on designing the eye-testing strategy algorithm.

- **Development**
  I carried out research on testing practices and then designed and implemented the actual eye testing logic. The testing strategy is based on the staircase method that terminates after some fixed number of reversal points and returns the estimated threshold value. This threshold value is calculated as an average of reversal points. As later suggested by the client, the logic was extended to tolerate one wrong answer as if it did not happen. All processing is done separately for each eye.
  Classes I have implemented:
  StaircaseEyeTestingStrategy, EyeTestingStrategy,
  EyeTestResult, Question, Symbol, UserAnswer

  Later on I added a chart generation to provide visual results of the test. According to our client's suggestion during our second meeting I extended this feature to generate PDF report with all the results. Here I faced interesting problem: generate charts and PDFs in memory as byte streams since we did not want to store any state on server for security reasons. Subsequent improvements were to parametrize the eye-testing strategy so that its initial configuration can be specified by TV operator. I ensured that the code is well commented, especially the eye-testing procedure. During the development I learnt about many new tools and libraries either through their usage (GitHub, PDFbox, JFreeChart, Salenium) or from other team members (e.g. about Jetty).

- **Testing**
  Before committing any change to our project I tested my module in separation. I used graphical output – plotting the progress of the eye-testing method as a way to test my code. (see Appendix A) These visual outputs were later used as a part of the PDF report with eye-test results. While testing, the inputs were randomly generated and special (boundary) cases were tested manually.

Close to the end when I finished my portion of work, Gabor and I started to work on automated integration testing. We decided to use a tool called Salenium WebDriver which I had to familiarize

with first. However, we did not have enough time to fully exploit this but it will be very useful for future projects.

Time invested per phase:

Design:	7 hours
Development:	25 hours
Testing:	6 hours
Client meetings:	3 hours
Deliverable reports:	5 hours

Contribution of other team members

Gabor Szarka
Gabor did the core server-side communication with TV and phone. He used Java's Jetty to set up WebSockets. Together with members of TV and Phone team they tackled with all the problems such as synchronisation issues and timeouts. He also implemented e-mail sending, audio generation and started to do integration testing. He also taught me some basics of GitHub as it was new to me.

Bradley Kemp
Bradley worked on the back-end component of the TV and implemented the WebSockets interface for communication with server as well as an API object that is to maintain the system's state and to handle and send requests. He also implemented APIPlayback tool to simulate all the API requests for testing the TV component. Later on he also implemented logic for soft timeouts.

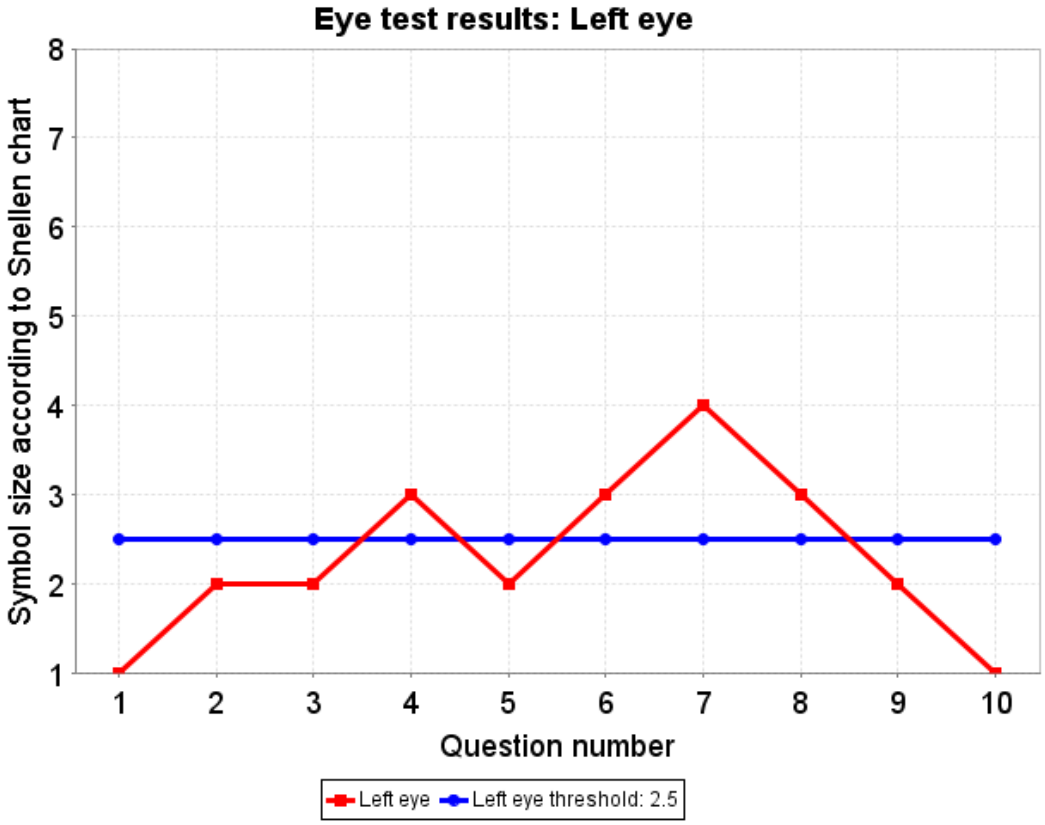
Alex Coplan
Alex did a lot of research about eye-testing symbols and their norms. He prepared special optotypes and developed the user interface of TV component. He also wrote the system for an operator to configure the TV before the test is started. He also set up the GitHub repository.

Coltman Jamie
Jamie and Ruby were working on Phone component which involved handling all the communication with the server during the test and interpretation of inputs from user. Jamie was new to JavaScript but he learnt it along the way with other team members' help.

Whipp Ruby
Ruby worked on Phone component. She was new to JavaScript as well but made a big progress and delivered her piece of work. She was also our team manager and made significant contributions to deliverables: Risk Analysis, Requirements & Specification and Final Report.

Everyone's enthusiastic attitude towards the project made it successful and interesting to work on. I am glad that I could have worked with such clever and collaborative people. I learnt a lot from them.

Appendix A: testing



Sample output from random testing.