Wearable Computing WiSe 06/07

Tutor: Hendrik Witt

Instructors:
Holger Kenn
Hendrik Witt

Problem Sheet 2

Submission Deadline: 03.12.2007

1 Problem

Implement an application for computer based visual acuity testing that uses Herman Snellen's chart developed in 1862. The Snellen chart is a chart for testing visual acuity. It consists of letters or numbers printed in lines of decreasing size which a patient is asked to read or identify at a fixed distance (see figure 1).

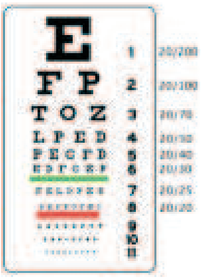

Figure 1: Snellen Chart

The application should support the following functionalities:

- Feature the possibility to run a visual acuity test on different distances to the screen and different screen sizes with different resolutions, e.g. TFT monitor and HMD.
- The letters that a test person has identified should be collected by the application from corresponding keys on the keyboard.
- The visual acuity of a "normal" person should be fixed to 20/20 with a visual angle of 5' of arc.
- After finishing the test, which is typically indicated by a test person not being able to correctly identify 1 or 2 letters in row, the application should present the results of the test. These are:
  1. achieved visual acuity and
  2. minimum font size (measured in millimeters) that is still perceivable by the person on a wearable user interface (focal distance 2 feet).

  Please do explain you calculations separately.

2 Points

2 Problem

Design an empirical user interface experiment that tests the impact of different properties on the perceptibility of presented text information on a wearable user interface with real users. The impact of the following aspects should be examined[1]:

- *Color*
  Color is mostly used in modern interfaces to optimize its look and feel and general design. Test reasonable combinations of different color pairs for background, foreground, and font color. Particularly include in your experiment combinations that are able to uncover red/green and blue/yellow weaknesses of test subjects.

- *Color Contrast*
  Color contrast is one important factor for readability. Change the color contrast to investigate how readability of text information is effected. There has been proposed a special algorithm by the W3C consortium to optimize color contrast for web interfaces[2] whose applicability should also be investigated for wearable user interfaces.

- *Font type*
  Different font types are widely used in print media but rarely applied in user interface design. Investigate how the use of different font types (e.g. fonts with and without serifs) and font faces (i.e. bold and italic) can support the readability of text information presented on an interface.

- *Light conditions*
  Light conditions change over time in wearable computing environments because wearable computing usually takes place in highly mobile and outdoor environments. Investigate the impact of changing light conditions on the readability of text information on a user interface by changing illumination of the environment.

All experiments should make use of the MicroOptical SV-6 monocular HMD as output display. As the underlying tool to conduct your study, use your own visual acuity programm as a starting point and add needed tests as well as logging functionality.

The experiment description must include at least the following aspect:

- Goal of the experiment
- Experiment setup
- Used method
- Metrics and calculations used
- Experiment hypothesis
- Schedule and working plan for user tests and evaluation of results

Make sure that the experiment description is self-contained and includes all information that would be needed by others to reproduce your experiment. The experiment design will lay the foundation to judge the usefulness and correctness of the planed experiment to be conducted afterwards.

8 Points

Note: The submission of problem sheets has to be done by e-mail to hwitt@tzi.de with the following subject:

[Lecture Wearable Computing] Problem Sheet No.{*number*}. If problems require the implementation of software it has to be submitted along with the problem sheet containing documented source code, compiled binaries, and 2 small scripts (for Windows and Linux) to run the application.

[1] One aspect per group only. Assignments will be done in the next tutorial session!

[2] see http://www.w3.org/TR/AERT#color checkpoint 2.2 for details on the algorithm