Lehrstuhl für Medieninformatik
Institut für Information und Medien, Sprache und Kultur
Universität Regensburg

Interaction Technology and Techniques
Assignment 11: Ubiquitous Computing

Summer semester 2015

Submission due: Tuesday, 31. July 2015, 23:55

Hand in in groups of max. two.

Your task is to design and implement a novel interaction technique within the Ubiquitous Computing paradigm and quantitatively compare its performance to a traditional interaction technique.

11.1: Read up on Ubiquitous Computing

Download and read the paper *"The Computer for the 21st Century"*[1] (Weiser, 1992). Read appropriate further literature.

Concisely answer the following questions:

- How do current mobile phones fit into Weiser's vision of Ubiquitous Computing?
- What are ethical implications of the Ubiquitous Computing paradigm?
- *"The future is already here — it's just not very evenly distributed."* (William Gibson). Describe two examples where UbiComp has already entered everyday life.

Hand in the following file:

ubicomp.txt: a plain-text file containing your answers

Points

- **2** Good answer to first question
- **2** Good answer to second question
- **2** Good answer to third question

11.2: Develop a novel interaction technique

Develop a novel interaction technique that significantly speeds up interaction with computers compared to the prevalent approach at the moment. The technique may apply to desktop UIs or smart environments but should encompass UbiComp aspects (e.g., multiple devices, tracking of users, …). You do not need to incorporate the WiiMote or PyQtGraph into your application (but it may help).

Follow these steps:

- Find a common human-computer interaction task that currently takes significant time (e.g., because it requires the user to remember something, switch between multiple applications, move around the room, etc.) and estimate or measure the required task completion time using the best available technique. The task may apply to desktop computers, mobile phones or other systems.

[1] www.ubiq.com/hypertext/weiser/SciAmDraft3.html

- Improve on the status quo by designing a novel interaction technique that *could* significantly reduce the task completion time. Document this interaction technique (short description, illustrations where necessary) and give an estimate of how much your approach could reduce task completion time (e.g. using KLM)
- Implement the novel interaction technique and optimize it (e.g., reduce latency and jitter for pointing)

Suggestions for topics:

- Speed up authentication by automatically logging the user into a computer once they approach it (see e.g. *"The Active Badge Location System"*[2] (Want et al., 1992)).
- Allow the user to zoom in or out of a map by leaning forwards or backwards (see e.g., Lean and zoom: proximity-aware user interface and content magnification[3] (Harrison & Dey, 2008))
- Allow users to copy text to the clipboard on one computer and paste it on another computer, whereby the clipboard contents should be coupled to each user (e.g., by identifying the user on both computers via their Bluetooth-enabled mobile phone or via an IR marker that is recognized by the WiiMote, etc.)

Write a Python application which demonstrates the novel interaction technique. Hand in the following files:

- **ubicomp.py**: a Python script that implements this application.
- additional files as needed for running the application

Points

- **1** The Python script has been submitted, is not empty, and does not print out error messages.
- **2** The application implements a UbiComp interaction technique for a common task.
- **2** The application is robust and error-tolerant.
- **2** The application reacts quickly and precisely to user input.
- **2** The user interface is intuitive and guides the user where necessary.
- **2** The script is well-structured and follows the Python style guide (PEP 8).

11.3: Evaluate and document your interaction technique

Compare your novel technique to the status quo in a quantitative use study with at least four participants. Take care to mitigate confounding variables. Conduct a simple statistical analysis (e.g., t-tests, ANOVA) to determine whether a statistically significant difference between your technique and the status quo exists. Try to be as thorough as possible. It does not matter whether you can actually show an improvement.

Write a short report (max. three pages) which includes:

- your names
- an illustration or screenshot of the application
- a description of usage scenario, status quo, and your novel interaction technique.
- a short description of implementation details
- a description of study setup and results, including statistical analysis and discussion of results

Furthermore, create a short demo video of your application. It should fulfil the following requirements:

- format: 720p, 25 or 30 fps, MP4 format
- max. 1 minute
- short title slide (see below)
- all commentary as subtitles, no voice-over narration

[2] http://alumni.media.mit.edu/~dmerrill/badge/Want92_ActiveBadge.pdf
[3] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.8843&rep=rep1&type=pdf

- no copyrighted material

The video should start with a short description of the problem that is solved, followed by a demonstration of the system. If you want, you can also include evaluation results.

Hand in the following files:

- **ubicomp-interaction.pdf**: a short report as described above
- **ubicomp-interaction.mp4**: a short video demonstrating your application

Points

- **1** The report is well-written.
- **1** The usage scenario is described/illustrated well.
- **1** The implementation is described in sufficient detail.
- **1** The quantitative evaluation is realistic and fair.
- **1** The quantitative evaluation is described in sufficient detail.
- **1** The results of the evaluation have been analyzed correctly.
- **1** The results of the evaluation are presented and discussed well.
- **1** The video is in the format defined above.
- **2** The video presents the concept well.
- **2** The video looks nice.

Submission

Submit via GRIPS until the deadline

All files should use UTF-8 encoding and Unix line breaks. Python files should use spaces instead of tabs.

Have Fun!