School of Electrical Engineering and Computing
SENG2260/SENG6260 – Human-Computer Interaction

Assignment 1: Low Fidelity Prototypes (20%)

Submit using Blackboard by **EOD Friday September 14th 2018**

Design problem

The following is the design problem that each group will undertake:

There is increasing interest in virtual and augmented reality display technologies and the immersive interactive environments that they enable. Although head-mounted display technology is not new, 2018-2019 will see an unprecedented release of VR/AR systems for mainstream use, e.g. Facebook's Oculus Rift, Microsoft's HoloLens, HTC/Valve's Vive, Samsung's VR Gear and Sony's Playstation VR. In addition to the challenges of engaging with a general user base, it is unclear what the VR/AR "killer app" will be. Previous research has explored military, medical and educational use of similar technology. However, as these advanced user interfaces enter mainstream usage there are exciting opportunities to explore new applications of this technology and consider how this may impact human-computer interaction (HCI)/user experience (UX) approaches to designing, prototyping and evaluating user interfaces.

The **group project** this year will consider the use of the Microsoft HoloLens[1] in a library. Your group must design the user interface for a library-based HoloLens interactive system.

Low Fidelity Prototyping

In this group assignment you will do your first prototype for your team project, which will be your interface design as a low fidelity prototype. This tends to be a paper prototype but could be cards, storyboards, low tech (e.g. cardboard) mock-up or a combination of these. The low fidelity prototype should be able to handle at least three use scenarios. These scenarios could be the scenarios you described in the task from lab 3.

Later labs offer you an opportunity to start building the low fidelity prototype. The lab and workshop time in week 7 is set aside to test the prototype on your classmates.

In the week 7 your participation in testing is required. Prior to this step, your team should have conducted a *heuristic evaluation* of your prototype and corrected any errors or examples of bad design that had arisen. This prevents you from wasting valuable user testing time discovering things that could have been discovered without involving users. This strategy is aimed at getting the most value out of your test users.

Choosing What to Prototype and Test

You may need to adjust your scenarios so that they explore the riskiest parts of your interface. A part of your interface is **risky** if its usability is hard to predict, or if its usability strongly affects the usability of the whole system. For example:

- **Novel design**: Parts of your interface that are new and different are potentially risky. By contrast, a username/password form is not risky because it is a familiar and well-understood idiom.

[1] See https://www.microsoft.com/en-us/hololens

- **Frequent use**: A frequently-used feature might be risky, because the efficiency of the whole interface depends strongly on it.
- **Error danger**: Any feature in which user errors might be common or hard to recover from is risky.
- **Complexity**: A complicated or subtle part of your system is risky. The configuration interface for a firewall, for example, would be risky because it involves system-level concepts that users may not know or understand. A font selection dialog might be risky simply because of the number of choices it offers.

Risky parts need the most design iteration, so they will give you the most payoffs from prototyping. In other words, do not waste your effort on prototyping a login screen, but do make sure to prototype a novel, complicated or frequently-used dialog box. Not every risky part can be easily tested with low fidelity prototyping, but if you make sure your scenarios cover the risky parts now, you will be able to plan your subsequent prototypes better.

Preparing for Testing

Before testing your prototype, you should:

- **Conduct a heuristic evaluation**. As noted above.
- **Build your prototype**. Draw the static background, menus, dialog boxes, virtual objects and interface components. Decide how to implement the dynamic parts of your interface. Hand-sketching is encouraged. You do not have to prepare every possible screen in advance; it may be much easier to write responses on the fly.
- **Prepare a briefing for test users**. This should be, at most, a page of information about the purpose of your application and any background information about the domain that may be needed by your test users (who will be SENG2260/SENG6260 classmates) to understand it. These are also your notes for the briefing, so make them short, simple and clear, not dense wordy paragraphs. This is not a manual or quick-reference card. It should not describe how to use the interface.
- **Write your 3 scenario tasks on separate index cards**. Just write the concrete goal(s) of the task (e.g. "buy milk, tomatoes, and bread"). Do not write the specific steps to follow, since that is for your users to figure out. The tasks should be brief, roughly a couple minutes to run.
- **For each scenario, determine what exactly you are looking for and how to determine if you have found it**. If you are measuring complexity, try timing the users (possibly against a time you think should be reasonable). If you are measuring ease of use, count the errors the user makes and how long it takes them to realize there is an error and what they should do about it. If people become really stuck and you have to prompt them, make a note of this.
- **Do not try to correct any mistakes on the fly** (even simple or trivial ones). Just note them and fix them in the next iteration.
- **Choose roles for your team members**. One person must play the computer (changing screen sketches or the like). The other team members will be observers. We will not bother with a facilitator for these pilot tests. It may be useful for you to swap roles after every user on Testing Day (week 7), so that each of you gets a chance to try each role, but decide how you will do it in advance (and document this in your minutes).

- **Practice running your low fidelity prototype**. Every team member should practice playing the computer, learning the steps involved in making the prototype functional, such as rearranging pieces and writing responses. It is not important to be fast, just competent and confident. A few trials are enough. Make sure your prototypes can handle the 3 scenario tasks in question.

Running the Tests

When you run your prototype on a user, you should do the following things:

- **Brief the user**. Use the briefing you wrote up to describe orally the purpose of the application and background information about the domain. Do not waste too much time on this: 1 minute should be enough.
- **Present one task**. Hand the index card to the user and let them read it. Make sure they understand the task.
- **Watch the user do the task**. Take notes from your observations.
- **Repeat with the other tasks**. Run as many tasks on the user as you have time for.

Bring extra materials for testing. Having extra blank Post-it notes, correction tape, and index cards on hand will help you improvise if a user does something unexpected, or help you make small fixes to your prototype between users.

Playing Test Users for Your Classmates

On Testing Day, when you are serving as a user, you should:

- **Relax and enjoy yourself**. You are not being tested – the interface is. Part of the point of this experience is to feel what it is like to be the user in a user test, so that you can empathize with them.
- **Be cooperative**. Do not be intentionally dense, e.g. looking for Exit everywhere but the File menu. Interact with the interface as you would if you were really using it.
- **Think aloud**. Help the observers understand what you're thinking by verbalizing your thought process. "Let's see, I want to enter this bottle of milk, so where's the scanner... oh, here it is. I'll scan the bottle like this, oops that didn't work, let me find the bar code..." You get the idea.

Written Report [total marks 20%]

You should hand in an electronic report in PDF format with the following parts:

- **Heuristic Assessment** [/15] List the heuristics you believe are most important for your interface. Describe how they were evaluated (walkthrough, best practice, …). List any problems that were discovered and how they were corrected before the prototype was presented to the users.
- **Risk assessment** [/10] List the parts of your interface that you consider risky and why, i.e. categorize the risks by either ordering from highest to lowest or add weights. State which scenarios will test

each one, and state how you plan to prototype them (paper, computer, or other) to mitigate the risk.

- **Briefing** [/5] The briefing you gave to users.
- **Scenario tasks** [/10] The tasks you gave to users, as you wrote them on the cards. Include screen shots or photos of your prototype. Show the prototype in interesting stages, not just a blank window.
- **Observations** [/30] Usability problems you discovered from the testing. Describe what users did, but do not record users' names. Record relevant observations such as
  - How long a task took.
  - Whether the users had trouble with a task (and why).
  - Any revealing comments made by the users. For this reason, testers should be encouraged to "talk to themselves" while using the prototype.
- **Risk resolution and prototype iteration** [/25] What did you learn about the risky parts of your interface from this prototype? Propose design solutions for the usability problems you found. Describe how your prototype will change between your Testing Day users and the real users.
- **Minutes and summary of meetings** [/5] Attach all minutes along with a summary of meetings. In the summary/table comment on attendance (especially if one group member is constantly absent or late), action lists and whether they were completed on time (if not, why not and what was done in response), major decisions made regarding the project and a commentary on group dynamics (is performance of all group members satisfactory?).

Deliverables:

The marking form for this assignment is available with this assignment specification on Blackboard.

The main deliverables of this assignment are:

- Low Fidelity Prototype Report (1 per group) to be submitted via Blackboard
  - DEADLINE: **EOD Friday September 14th 2018**

Reports should be submitted as **PDF documents**. There is no target page length. Quality, criteria coverage and team effort are better than quantity. FYI two HD submissions from last year were 31pgs and 80pgs.

Late Submission:

Assignments submitted after the deadline will have marks reduced by 10% per day late. (The weekend counts as 2 days.) For example: An assignment worth 20% marked at 78%

• On time:	0.78 *1.00 * 20 = final mark = 15.6
• 1 day late:	0.78 * 0.90 * 20 = final mark = 14.04
• 3 days late:	0.78 * 0.70 * 20 = final mark = 10.92
• 5 days late	0.78 * 0.50 * 20 = final mark = 7.8