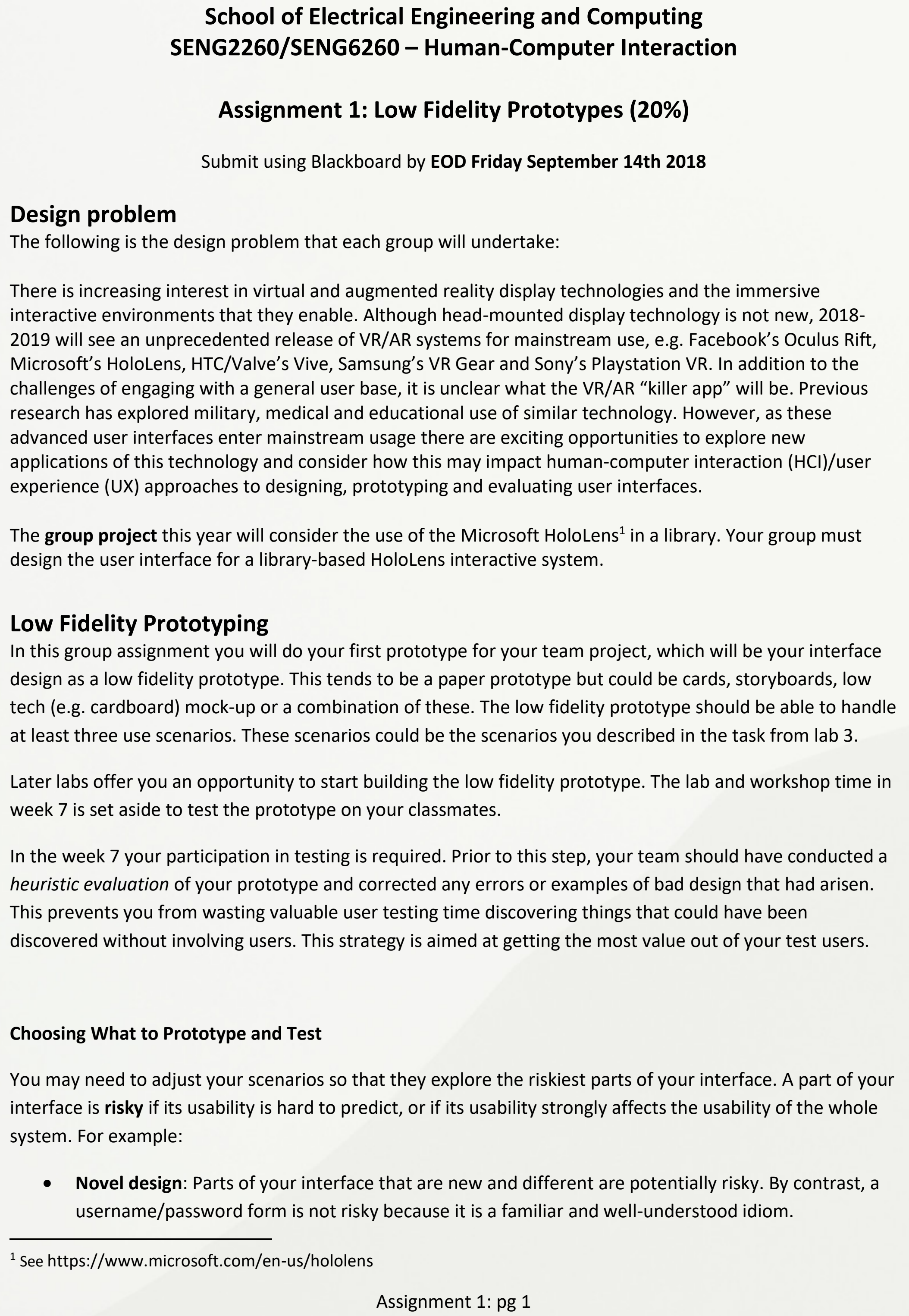
An Evaluation of Augmented Reality in a Library Using a Microsoft HoloLens SENG2260 Assignment 1
My Dudes

Ethan Bentley c3130282@uon.edu.au
Jasbir Shah c3182837@uon.edu.au
Jay Rovacsek c3146220@uon.edu.au
Josh Brown c3283797@uon.edu.au
Dean Morton c3252227@uon.edu.au
Tom Milton c3261280@uon.edu.au
Ryan Lengling c3282142@uon.edu.au
Elijah Hunt c3236563@uon.edu.au
Benjamin Walters c3207457@uon.edu.au

September 16, 2018

Contents

1 Assignment Requirements



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).

School of Electrical Engineering and Computing

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

SENG2260: Human-Computer Interaction

Assignment 1: Low Fidelity Prototyping (20%)

DEADLINE: EOD Friday September 14th 2018

Group name:___________________________________

PROTOTYPING REPORT	POOR	BELOW AVERAGE	AVERAGE	GOOD	EXCELLENT	Mark
Heuristic Assessment /15	0-3. Limited or no evidence of heuristic assessment. Ad hoc tasks and/or changes applied. Limited evidence of evaluator involvement.	4-6. Some heuristics applied. Limited scope/poor choice of heuristics. Unrealistic tasks/solutions. Unstructured evaluator participation.	7-9. Evidence of heuristic assessment. Limited evaluation scope/problems found. Evidence of evaluator participation. Mainly descriptive review.	10-12. Evidence of good heuristic assessment. May be limited in scope but well-structured and detailed. Evidence of problem/solution review and evaluator use.	13-15. Comprehensive choice and use of appropriate heuristics. Problems discovered and solutions discussed. Reflective review of assessment.	
Risk Assessment /10	0-2. No or very limited risks identified. Unstructured assessment. No or poor mitigation plan.	3-4. Some risks identified. Limited in scope and/or unrealistic risk scenarios. Brief or limited mitigation plan.	5-6. Evidence of risk assessment. Limited scope or overly basic risk scenarios. Adequate, mainly descriptive mitigation plan.	7-8. Evidence of good risk assessment. Well-structured and detailed. Good risk scenarios supporting realistic mitigation plan.	9-10. Excellent risk assessment with comprehensive risk scenarios and mitigation strategies. Rationale for risk ranking.	
Briefing /5	0-1. No or very limited briefing. Lack of appropriate test information.	2. Poorly structured or overly long brief. Briefing describes how to use the interface or several issues across scope, formatting, level of detail.	3. Adequate briefing document. More than one minor issues with scope, formatting or level of detail.	4. Good briefing document. Clear user instructions. Well formatted with only one minor issue in scope, layout or level of detail.	5. Excellent one page detailed briefing. Appropriate information on background and application purpose.	
Scenario Tasks /10	0-2. No or limited description of scenarios. Limited or unrealistic tasks. Limited relevance for use in prototype. Brief description.	3-4. Poor or overly basic tasks. Limited scenarios. One or more of limited value for prototype improvement, poorly presented, incomplete.	5-6. Details of 3 adequate scenarios tested. Tasks have measurable outcomes. Screenshots/photos of prototype.	7-8. Three good scenarios. Interesting and representative tasks. Well structured descriptions with photos/screenshots	9-10.Excellent representative scenarios with innovative tasks. Excellent description in context of prototype.	
Observations /30	0-6. No or very basic observations. Poor quality/irrelevant problems identified and limited engagement with task. Overly brief and/or poorly structured.	7-12. Basic observations. Limited problems identified. More than one issue from: participants identified, limited detail, poorly presented, incomplete.	13-18. Descriptive overview of observations. Some exemplars of relevant observations and problems identified. Minor issues in level of detail or relevance.	19-24. Structured overview of observations and problems. Detailed exemplars of relevant observation. Reflective discussion.	25-30. Excellent reflective report on usability observations. Comprehensive, detailed and consistent review of testing and significance.	

2 Project Outline

3 Heuristic Evaluations

3.1 Heuristic Assessment

Through the design process we evaluated our user interface using "Jakob Nielsen's 10 principles for interaction design".
These heuristics allowed us to make informed decisions when creating and making iterations to our prototype.

3.1.1 Visibility of System Status

Our prototype gave user feedback by quickly transitioning between slides (screens) based on the user's gestures. When a user performed a certain gesture it was established that the next slide would be displayed within reasonable time, based on the available performance metrics of the Microsoft Hololens.

This type of feedback would allow users to interpret if they had made an error as well as having a 'screen shake' effect if the user tried to input a gesture that was not recognised. In the tutorial scenario, if users input a gesture incorrectly then they were given feedback and also shown with an animation of an avatar representing a human hand, how the gesture was to be performed.

3.1.2 Match Between System and the Real World

The system prototype used a WIMP based metaphor UI display with simple icons and folders to allow users to quickly transition from other computer based systems that use a mouse and keyboard input and mobile based systems which use touch based interfaces - allowing them to quickly adapt to the Hololens gesture input methods.

3.1.3 User Control and Freedom

The bloom gesture which we associated with the heuristic for ease of use was used as an 'emergency exit'. If the user became stuck at any point they could use this gesture to return to the main menu. Allowing the user to feel free to explore the menus of the system without the worry of getting lost.

3.1.4 Consistency and Standards

Consistency was kept in regards to the types of icons displayed and the graphic layout of the icons. This was important as the system transitions between screens frequently and the layout of the icons and menus needs to be logical so as not to confuse the user.

The type of error feedback given when a user made an error was also kept consistent in all areas of the system.

3.1.5 Error Prevention

A 'screen shake' effect was provided to users as feedback when they attempted to use a gesture incorrectly or when data was input incorrectly. The tutorial scenario was created in an attempt to prevent users from performing gestures incorrectly. This tutorial could be accessed by users at any point if they needed to refresh their memory of what gestures could be performed to perform certain inputs.

3.1.6 Recognition Rather Than Recall

To decrease the memory load on the user the screens were designed to be minimalist in regards to the amount of information that was displayed per screen. Used in conjunction with the consistency of the icons and layout format of the screens the user was viewing, the user was able to recognize what screen they were on. However, issues presented themselves when the user was required to recall what step of the process they were up to.

Users frequently checked back to the task checklist while conducting the study, more feedback showing what menu they were currently in and feedback as to what step of the task they were currently on could have helped remedy this.

3.1.7 Flexibility and Efficiency of Use

Simple hand gestures that the user can quickly learn were used to increase efficiency. This was tested in our Tutorial scenario where we taught new users the commonly used hand gestures to see if they were able to navigate the system. It was decided to keep the amount of gestures low as to not confuse the user. When users started becoming familiar with gestures their speed at completing tasks increased.

3.1.8 Aesthetic and Minimalist Design

The slides which represented different screens being presented to the user only showed relevant information according to which screen was displayed. Allowing the user to easily transition between screens and quickly interpret what information was being displayed. The 'exit group' icon was not clear as many users found it hard to find. This could be fixed by making the button larger and separated from the bar.

3.1.9 Help Users Recognize, Diagnose and Recover from Errors

In the first iteration of the prototype, the error screen contained a webdings type that was to simulate an error message on the error screen. This confused users more as they were not able to recognize that it was an error message and drew the users attention away from the fact that they were being presented with an error screen. This was fixed in future iterations to make the error screen clearer.

3.1.10 Help and Documentation

The system provides its own tutorial when a new user is recognized. This tutorial is also available at any point from the main menu screen which can be accessed at any time using the bloom gesture.

4 Risk Assessment

5 Test User Briefing

6 Team Member Roles

7 Scenario Tasks

8 Observations

8.1 Tutorial Use Case

Tester 1		
Tasks	**Observation**	**Task Duration**
Open/Close the menu	Was confused, needed help	Overtime
Select	Completed with ease	Reasonable
Left/Right	Did incorrectly twice, needed help	Overtime
Back/Forward	Completed with ease	Reasonable
Additional Comments	Tester Reported Error page was unclear, and wasn't sure how to proceed. Bloom Gesture was unclea	

9 Risk Resolution and Prototype Iteration

10 Minutes and Summary of Meetings

References

[1] Place holder.com

11 Appendix