Course Project
TEXT ENTRY ON SMALL DEVICES

INFO 463: Input & Interaction

PURPOSE

The purpose of the course project is to give you an opportunity to leverage what you have learned in this class in the creation of an input technique of your own. You can reflect on what you have seen: input and interaction techniques, models of human behavior, underlying principles of human perception and motor control, and the clever design ideas that make a technique successful. You will bring all of this to bear in the development of your own interactive prototype demonstrating how to input text on small devices. You will be designing techniques that allow users to effectively input text on a small device such as a smartwatch, digital music player, fitness tracker, *etc*. For the purposes of this assignment, a smartphone is not considered a small enough device, although you may use a smartphone to simulate an even smaller device on its screen as part of your prototype.

GROUP SIZE

This final project is to be done **in groups of four (4) people**. You will have time to form your groups during class.

READINGS

Familiarity with the following resources is necessary background for your project; you should skim them before you begin. Together, they constitute the intellectual backdrop for this project, and ought to be very helpful to you. The articles are:

- MacKenzie, I.S. and Zhang, S.X. (1999). The design and evaluation of a high-performance soft keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99)*. Pittsburgh, Pennsylvania (May 15-20, 1999). New York: ACM Press, pp. 25-31.
- Zhai, S. and Kristensson, P.O. (2003). Shorthand writing on stylus keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '03)*. Fort Lauderdale, Florida (April 5-10, 2003). New York: ACM Press, pp. 97-104.
- Wobbrock, J.0., Myers, B.A., and Kembel, J.A. (2003). EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '03)*. Vancouver, Canada (November 2-5, 2003). New York: ACM Press, pp. 61-70.
- MacKenzie, I.S. (2007). Evaluation of text entry techniques. Chapter 4 in *Text Entry Systems: Mobility, Accessibility, Universality*, I.S. MacKenzie and K. Tanaka-Ishii (eds.). San Francisco: Morgan Kaufmann, pp. 75-102.
- Wobbrock, J.O. (2007). Measures of text entry performance. Chapter 3 in *Text Entry Systems: Mobility, Accessibility, Universality*, I.S. MacKenzie and K. Tanaka-Ishii (eds.). San Francisco: Morgan Kaufmann, pp. 47-74.
- Kane, S.K., Bigham, J.P., and Wobbrock, J.O. (2008). Slide rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. *Proceedings of the ACM Conference on Computers and Accessibility (ASSETS '08)*. Halifax, Nova Scotia (October 13-15, 2008). New York: ACM Press, pp. 73-80.

- Baudisch, P. and Chu, Gerry. (2009). Back-of-device interaction allows creating very small touch devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09).* Boston, Massachusetts, (April 4-9, 2009). New York: ACM Press, pp. 1923-1932.
- Schwarz, J., Harrison, C., Hudson, S., and Mankoff, J. (2010). Cord input: An intuitive high-accuracy, multi-degree-of-freedom input method for mobile devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '10)*. New York: ACM Press, pp. 1657-1660.
- Oney, S., Harrison, C., Ogan, A., and Wiese, J. (2013). ZoomBoard: A diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '13)*. Paris, France (April 27 – May 2, 2013). New York: ACM Press, pp. 2799-2802.
- Kienzle, W. and Hinckley, K. (2013). Writing handwritten messages on a small touchscreen. *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '13)*. Munich, Germany (August 27-30, 2013). New York: ACM Press, pp. 179-182.
- Dunlop, M.D., Komninos, A., and Durga, N. (2014). Towards high quality text entry on smartwatches. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '14)*. Toronto, Canada (April 26 – May 1). New York: ACM Press, pp. 2365-2370.
- Xiao, R., Laput, G., and Harrison, C. (2014). Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt, and click. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '14)*. Toronto, Canada (April 26 – May 1). New York: ACM Press, pp. 193-196.
- Chen, X.A., Grossman, T., and Fitzmaurice, G. (2014). Swipeboard: A text entry technique for ultra-small devices that supports novice to expert transitions. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '14)*. Honolulu, Hawaii (October 5-8, 2014). New York: ACM Press, pp. 615-620.
- Leiva, L.A., Sahami, A., Catala, A., Henze, N., and Schmidt, A. (2015). Text entry on tiny QWERTY soft keyboards. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '15)*. Seoul, South Korea, (April 18-23, 2015). New York: ACM Press, 669-678.
- Esteves, A., Velloso, E., Bulling, A., and Gellersen, H. (2015). Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. *Proceedings of the ACM Symposium on User Interface and System Technology (UIST '15)*. Charlotte, North Carolina (November 8-11, 2015). New York: ACM Press, pp. 457-466.

WHAT YOU'LL DO

Entering text on a computing device is a fundamental tasks users perform on a daily basis. As computing devices get smaller, so does the area available for text entry. Your objective is to design a new text entry method that is effective for small devices. An effective solution will: (1) make every lowercase letter of the alphabet available for entry (capitalization, numerals, punctuation, and special characters are not required, but of course, you can add those, too, if you wish); (2) afford users the ability to correct errors (a backspace feature is required, and beyond that, is up to you); (3) provide feedback conveying what text has been entered; and (4) require little training by the user to operate (no Morse code!). Although a valid design space in general, for this class, methods leveraging speech or voice recognition are <u>not</u> allowed.

You can select any device to implement your text entry technique. You can also simulate your device on other devices (e.g., showing a smartwatch face on a smartphone screen). You can decide to use existing devices with their respective capabilities *as-is* (e.g., the touchscreen on an Apple Watch, or the button on a Fitbit Charge), or you can envision *new* capabilities for existing devices (e.g., an Apple Watch with pan, twist, tilt, and click capabilities[1], or a Fitbit Charge with back of device interaction capabilities[2]). Whatever you decide, the assumed capabilities of the hardware must be clearly explained. Any proposed novel device capabilities must be supported by existing research and referenced accordingly. This limitation is in place to encourage exploration of other interaction techniques that, although created for a different purpose, may prove beneficial for text entry on small devices. Think outside the box, but no mind-reading smartphones!

The set of readings above are designed to get you started on understanding the current landscape for text entry on small devices. These readings contain some design ideas that might be applicable to your project and may be sources of inspiration for your work. Of course, in any work you do, if prior work inspires or informs you, you must cite it appropriately. How and whether you make use of these readings is up to you, but you ignore them at your peril!

Step 1. Define your goals, brainstorm, and sketch ideas

After completing the required readings, your group's first step is to develop **at least ten (10) design sketches of possible solutions**. These can be entirely new or *based* on design ideas from the readings. If they are based on those in the readings, they must be adapted or extended. Your sketches can be done in a tool like Microsoft PowerPoint or Adobe Illustrator, or just using pencil-and-paper (but you will have to scan these in later for electronic submission). They should each have a short caption (3-4 sentences) saying how the technique would work, and its benefits and drawbacks. The sketches are intended to help your group brainstorm, so use them to think broadly about how to allow users to input text on small devices.

Step 2. Refine your ideas

Next, take **the two (2) best ideas** your group came up with and create flipbooks for them. A flipbook is a sequence of sketches that shows an unfolding interaction sequence. You can do this by making a series of images, or by using a tool like Microsoft PowerPoint and flipping through slides. You could also use Adobe Flash to create an animation that is *not* interactive. The flipbook should show how the interaction works in detail. A paragraph description should accompany each of the two flipbook designs.

Step 3. Create an interactive prototype

Next, take **the one (1) best idea** and create an *interactive* prototype. An interactive prototype is an application that allows a user to experience the design directly using an input device (*e.g.,* mouse, finger on touch screen, *etc.*). When it is fully created, a user should be able to input text on your system. You do not need to build a full-fledged application or demonstrate your technique working on whatever hardware platform your team has chosen. (Again, you can simulate your device on a smartphone or a desktop computer. Physical buttons can be represented as virtual buttons, with your device drawn in pixels on the screen.) Your prototype, however, must be interactive, and any

---

[1] Xiao, R., Laput, G., and Harrison, C. (2014). Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt, and click. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '14)*. Toronto, Canada (April 26 – May 1). New York: ACM Press, pp. 193-196.

[2] Baudisch, P. and Chu, Gerry. (2009). Back-of-device interaction allows creating very small touch devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09).* Boston, Massachusetts, (April 4-9, 2009). New York: ACM Press, pp. 1923-1932.

functionality you presume your device has that is relevant to a text entry task must be replicated in a way that is usable by the end user.

You can build your prototype in any language you like. If your program runs in a web browser, please specify the desired browser and platform. You might consider using Adobe Flash / ActionScript or Microsoft Silverlight with VB .NET or C#. You can also use Java or JavaFX. Or you can use any native browser-compatible technologies (*e.g.*, HTML/CSS and JavaScript). It is up to you, but your prototype should be stable and usable by anyone who might like to try it. It should not only be bug-free, it should "feel right" to human users!

Step 4. Test and evaluate your interaction technique

Finally, you will run a short informal **user study with three (3) users** who are not members of your project team. They can be friends, family, students, or otherwise, but **they must not be members of this class.** In order to evaluate your prototype, your system must be able to present phrases to the participant, and the participant must be able to enter text into the system and see what text has been entered.

For your study, you will ask each participant to use your prototype to transcribe the presented phrase. After a phrase has been transcribed, a new phrase should be presented to the participant. Each participant will complete 45 text entry trials, with the first 5 files being discarded as practice. Your system should use the MacKenzie and Soukoreff phrase set found here: http://www.yorku.ca/mack/chi03b.html.

If upon starting the study, you find that your subject is unclear about what to do, stop the study, instruct them again, and restart only when you are confident that he or she can perform the tasks. Otherwise, your results will be confounded by your participant's confusion, and measuring confusion is not the point of the study!

Each participant will have to balance speed and accuracy. Tell them a good rule of thumb is to "proceed quickly and accurately," trying to balance the two concepts evenly. Proceeding recklessly is not balancing speed and accuracy, but neither is fixing every single error, especially errors that are not noticed until the participant is much farther along in the phrase. Make sure your participants indeed exhibit a balanced approach to speed and accuracy, trying hard to do both well but not sacrificing one for the other.

You must compute text entry performance metrics to evaluate the effectiveness of your prototype. To do so, you need to create a testbed that will capture the text input by the user. Download ***TextTest*** and ***StreamAnalyzer*** from http://depts.washington.edu/ewrite/eval.html. These programs were written by Dr. Jacob Wobbrock to enable text entry evaluations. The software runs on Microsoft Windows .NET framework. (Any up-to-date machine should have this already installed on it.) TextTest is an example of a text entry evaluation testbed. Spend some time interacting with TextTest to see how it works. Your testbed should behave similarly to TextTest, but you do not need to replicate all its functionality. Remember, your testbed should be able to: (1) present a phrase to the user; (2) allow the user to input text; (3) allow the user to see what text has been entered; and (4) present a new phrase once the user has finished transcribing the current phrase.
In addition to the functionality described above, your testbed must also create XML log files similar to those created by TextTest. These log files are required so that they can be parsed by

StreamAnalyzer. You can play around with TextTest and create log files of your own to see how they are formatted, but we will give you an example of a pared down version of the log file as part of the project assignment. Make sure the log files produced by your testbed can be read and parsed by StreamAnalyzer.

There are four parts to a StreamAnalyzer output TXT file. The top is the aggregate trial measures in space-delimited columns. Column headers also appear for you to paste into the top row of an Excel spreadsheet. (Use Freeze Panes to freeze the top row in place.) The column definitions themselves can be found in the Columns.txt file that accompanies the TextTest application. Read the definition for each column so you know what it means.

In the parsed results TXT file, you'll find the aggregate trial-level results at the top, with one row per trial. Below these, you'll find two character-level count tables, and two character-level confusion matrices. **You can ignore all of the character-level results for this assignment**. You just need to use the trial-level aggregate results at the top of the TXT file. We will provide more detail on how to handle the data produced by StreamAnalyzer in the **Evaluation Lab** taking place on Monday, April 24th.

CHECKPOINT SUBMISSIONS

Ten design sketches

There will be two checkpoint submissions. First, on Monday, May 1st, you will turn in your **ten (10) design sketches** during the lab session. These will not be graded at this time, but they must be complete. If they are not complete and satisfactory at this time, -10% will be deducted from your final project grade. Each group will present a description of their design sketches. This presentation will serve as an opportunity for teams to receive feedback and discuss their design decisions to date. You should scan any hand-drawn sketches so that you can put images on PowerPoint slides for public display. Each group will have five minutes to present their design sketches. Due to the time constraint, it would be unwise to attempt to show all of your design sketches. Instead, your group should pick two of the most promising sketches and create a presentation discussing those ideas.

Flipbook feedback and status update

Second, on Monday, May 22nd, you will turn in your **two (2) flipbooks** for your two best design ideas. These will not be graded at this time, but they must be complete. If they are not satisfactory at this time, -10% will be deducted from your final project grade. In addition to turning in your flipbooks, each group will present the status of their design process to-date. Each group will present slides discussing progress so far: design decisions made, designs the group is currently working on (by now, you should have narrowed your designs to the ones you will prototype and test); and your plan for the remaining weeks in the course. Your slides should show design ideas and iterations. If you already have a working demo, you can capture that demo as a video and play it from within your slides. Each group will have five minutes to present their status update. This presentation is meant **to help you!** Think carefully and honestly about how much work your group has completed so far and how much the group has to complete before the end of the course.

Turning in slides, sketches, and flipbooks

Slides for the two presentations must be sent through email to Abdullah Ali (xyleques@uw.edu) at least 24 hours before the start of the lab sessions. No late slides will be accepted. Your design sketches and flipbooks must also be emailed to Abdullah before the start of the lab sessions in which they're due. The design sketches and flipbooks do not need to be in a specific format, but they must meet the descriptions mentioned above (**Step 1** and **Step 2** under **WHAT YOU'LL DO**.)

A WORD OF WISDOM

Start now! Start now! Start now! Design and prototyping are not things that can be accomplished the night before a project is due. As the IDEO design firm touts, "Fail often to succeed sooner," and "Enlightened trial-and-error succeeds over the planning of the lone genius." You can't fail often or have enlightened trial-and-error unless you start with enough time to do so. Also, you don't want to simply jump onto the first seemingly good idea that you have, because often, the best ideas are the ones you reach after trying the first obvious ones. To be a successful designer, a project must consume you, occupying your thoughts while you eat, on the bus, and as you try to fall asleep at night… Dive in and get started!

DEMO + PRESENTATIONS

During the final exam period at **2:30 pm on Thursday, June 8**, each group will have 12 minutes to present their course projects. In your presentation, you must address the following: (1) the device your technique was built for and what capabilities you presume the device to have; (2) the design and key concept behind your text entry technique; and (3) the results from your text entry evaluation. In addition to the group presentation, your group will also demo your interaction technique. If your technique can run on a Windows PC we can project it for the class. If your demo runs on a mobile device, we will pass the device around the class (first to the instructors.) If the demo is on a mobile device, it would be preferable to have the demo on more than one device so they can demo on one device and pass around the others. Each group will have 12 minute for their presentation + demo. The presentation slides must be done in PowerPoint and sent to Abdullah Ali by **11:59 pm on Wednesday, June 7**.

WHAT TO TURN IN

At 2:30 pm on Thursday, June 8, each group should turn in the following as a cohesive report:

1. Design sketches and early design concept descriptions
2. Flipbook designs with descriptions
3. Interactive prototype and README for running and using (if additional emulation software is required to run the prototype, include this software or easy download links & instructions)
4. User test results (1 page write-up)
5. Slide deck used in the final presentation
6. *Optional*. Any other materials (*e.g.,* videos)