Interactive Prototype and Usability Testing Plans

of CSC318 Group 3

B.Hu - benj.hu@mail.utoronto.ca

D.Cho - dawing.cho@mail.utoronto.ca

R.Qiu - rex.qiu@mail.utoronto.ca

P.Bilodeau - philip.bilodeau@mail.utoronto.ca

M.Huang-Hobbs - maxwell.huang.hobbs@mail.utoronto.ca

TA: Eugene Cheung

March 18, 2015

PART I: Evaluation of individual designs

Feedback (Solutions/features that we liked and agreed on)

- Help exists already but it is not very helpful
- Should be integrated in shell or a standalone native application
    - Node-webkit allows web-apps to run native
- “helpbot” or generally more direction for the user separate user input from shell output (readability)
- Simple indication of successful command execution / unsuccessful command
- Visual aid for file types
- File sidebar: highlight on file creation, deletion, when asking for rm confirmation. (better “spatial awareness” with respect to the location in the file system the user is in / scope [current state])
- rm, chmod confirmation text
- Input ‘grabbing’ (back and forth from a program)
    - Plays neatly into the ‘chat’ metaphor
    - Similar to confirm messages
- Auto-completion, correction, next-command prediction
- Highlight affected files when file globbing within a directory
- Some kind of visual tutorial for the interface (a la slack’s initial user experience)
- Tooltips (help-text when typing commands)
- Command searching by tags
- Syntax coloring

PART II: Create a prototype for a combined solution

An interactive web-based CLI that shows some (faked) functionality:
https://github.com/Adjective-Object/simple-term-mockup
(Attached on Portal is the source code as well, in a .zip file)

An Invision Prototype that displays some of the features and how they work:
http://invis.io/S52GSQNV9

PART III: Interaction sequences

Sequence 1:

0. Open Command Line Interface
1. Determine task and the required commands needed to complete it (i.e., pull from a Git repo).
2. Enter commands repeatedly
    a. using the search feature, look up commands and their arguments/syntax
    b. utilize autocomplete for frequently used commands (i.e., pull)
    c. look through sidebar to keep track of scope, as well as the files in the working folder
    d. consult the help pages for more specific information
3. Complete task
4. Close Command Line Interface

Sequence 2:

0. Open Command Line Interface
1. Determine task and the required commands needed to complete it (i.e., remove a file).
2. Enter commands repeatedly
    a. Utilize features if necessary (speeds up process)
3. Complete task
4. Close Command Line Interface

Users who are unfamiliar to the command line or given task will be able to reach their goal with the help of all the tools and features described within Sequence 1. However, if a user has previous experience with a given task, he may forgo making use of any of the a-d substeps and complete his task unhindered. In this way, our system is flexible in that it provides tools which cater to users who need them, but it does not slow down users who know what they are doing.

PART IV: Usability testing plans

Research Protocol

1. **Project Title:** Interviews and Observations of Command Line Interface User Experience.

2. **Investigators:**

B.Hu - benj.hu@mail.utoronto.ca

D.Cho - dawing.cho@mail.utoronto.ca

R.Qiu - rex.qiu@mail.utoronto.ca

P.Bilodeau - philip.bilodeau@mail.utoronto.ca

M.Huang-Hobbs - maxwell.huang.hobbs@mail.utoronto.ca

3. **Purpose:** The purpose of our research is to gather opinions and feedback on our prototype, with regards to the added features and general usability of our redesigned command line interface A brief description of our design concept is a CLI assistance software which provides visual cues and intuitive features to help usability.

4. **Process to be followed:** We will brief the participants about the purpose of the study, explain the consent form to them, and ensure that they sign the consent form. We will then engage the participants in observed command line usage tests with our prototype, short follow-up questionnaires, and opinion based interview questions. The workspace will be in the CSC318 Lecture room.

5. **Participant selection:** Participants will be chosen from individuals present on the test day in the lecture hall. In general, they will be characterized by age, program of study, and level of previous experience with command line interfaces.

6. **Relationships:** Our relationship to the participants may be described as follows: No relationship.

7. **Risk and benefit:** There will be minimal risk to the participants, for example that they will only feel that they have wasted their time. The only benefit will be to contribute to the education of the investigators. Participants are free to withdraw before or at any time during the study without the need to give any explanation.

8. **Consent details:** We will brief the participants about the purpose of the study, and explain the attached consent form to them, and ensure that they consent to participate and sign the consent form.

9. **Compensation:** Participants will receive our sincere gratitude.

10. **Information sought:** The information to be sought is described in the attached questionnaire, interview script, or observation protocol.

11. **Confidentiality:** Information will be kept confidential by the investigators. Names or other identifying or identified information will not be kept with the data. The only other use will be to include excerpts or copies in the assignment submitted, but names and other identifying or identified information will not be submitted

Consent Form: Command Line Interface User Experience

I hereby consent to participate in a research study conducted by B.Hu, D.Cho, R.Qiu, P.Bilodeau and M.Huang-Hobbs for an assignment in *University of Toronto Computer Science 318, Design of Interactive Computational Media*.

I agree to participate in this study with the purpose of testing our command line interface prototype.

I understand that

- The procedures to be used are: observation, questionnaire and interview.
- I will receive no compensation for my participation in this study.
- I am free to withdraw before or any time during the study without the need to give any explanation.
- All materials and results will be kept confidential, and, in particular, that my name and any identifying or identified information will not be associated with the data.

PARTICIPANT

Name (please print) _____

Signature _____

Toronto, Date _____

INVESTIGATOR(s) Name _____

Signature _____

Research Instruments

Participant background information

1. Do you know what a command line interface is (CLI)?
2. Are you currently using a command line interface in any capacity?
3. How long have you been using a command line for if you have used one before?
4. How often do you use a CLI on average?

Inquiry about current practices in the problem space

1. What was the hardest part about using the CLIs?
2. What is your general opinion regarding CLIs?
3. Are there tasks in particular do you find difficult, annoying, or inefficient to accomplish using a CLI? (What would you change about it?)
4. Now if I'm a newbie student and I want to learn CLIs, what advice would you give me?
5. In which cases would you choose to use a CLI over a GUI given the option to in your day-to-day life?
6. Are there specific use-cases which you would recommend learning a CLI over another?
7. What do you think the benefits of using a CLI are?
8. What did/do you do when you do not know what command to use to complete a given task?

Task Performance assessment

1. find your current working directory's contents
2. use help feature to lookup some commands (e.g. ls)
3. input some command correctly
4. determine which things in the current working directory are folders

Follow up questions & questionnaire

1. What are your opinions towards the new features added and the overall design of the Command Line Interface? (Best feature/worst)
2. Do you think existing command line interfaces have problems?

3. Do you find these proposed solutions effective in solving existing problems with Command Line Interfaces?
4. Would you use this over a regular CLI? Why or why not?
5. Are there any specific improvements that could be made to the design/features?

Overall Reaction

Terrible					Wonderful
Difficult					Easy
Frustrating					Satisfying
Inadequate Power					Adequate Power
Dull					Stimulating

Screen and Appearance

Reading the Screen	Hard					Easy
Organization of Information	Confusing					Simple

Terminology and system information

Use of terms in the system	Inconsistent					Consistent
Is terminology related to task?	Never					Always
Position of message on screen	Inconsistent					Consistent
Prompts for input	Confusing					Clear
Error Messages	Unhelpful					Helpful

Learning

Exploring new features by trial and error	Difficult					Easy
Learning to operate the system	Difficult					Easy

Remembering names and use of commands	Difficult					Easy
Performing tasks is straightforward	Disagree					Agree

Interface Capabilities

Speed	Slow					Fast
Reliability	Unreliable					Reliable
Mistake Correction	Difficult					Easy
Designed for all levels of users	Disagree					Agree

What is your general impression of this Command Line Interface?

What are the most negative things you have to say about this Command Line Interface?

What are the most positive things you have to say about this Command Line Interface?