

**CMSI 370-01**  
**INTERACTION DESIGN**  
Fall 2015

## Assignment I020 (due I022) Feedback

All applicable outcomes can now reach maximum proficiency values with this assignment.

Dustin Kane

ranneyd / dustinpkane@gmail.com

*Notes while reading:*

- Fun trick with the live cursor there :) +(2a)
- In the *Typical Behavior* section, there is a presumption that the command loop applies solely to operating system commands. Not so—you issue commands to games too (e.g., /dance or other directives inside game chats; the classic Infocom text adventure games). Plus, some might argue that the Google search field is also a form of command interaction, with its ability to do more than just conduct search terms. REPLs are arguably command-loop interfaces as well, with commands that do not necessarily go to the operating system. (1a, 2a)
- That said, the good news is that the rest of the content is [mostly] not specifically bound to operating system interaction, and remains applicable to other environments that accept commands. The areas the content would have been affected would be *Component in Action* (showing more than just the operating system command prompt) and *Variants*. (1a, 2a)
- The provided *Component in Action* video is great, although could have used some commentary describing what is going on in the screengrab sequence. (2a)
- The *Priority Metrics* section is very cogent. It reads almost exactly like Shneiderman's own discussion of the command interaction style. Check it out in *Designing the User Interface* if you haven't already. +(1b, 2b)
- In terms of learnability, one insight that I was hoping you would spot is the way the command line interaction style is the direct converse of menus/forms/dialogs: instead of relying on *recognition*, it relies on *recall*, and that is why it does poorly with learnability. In that sense, tab completion is not only an efficiency aid, but a learnability one, because it shows the user what is available given the typed text so far. This shifts some cognitive load from recall to recognition. However, as you said, *discovering* tab completion is a whole other learnability issue in itself. (1b, 2b)
- The *Key Characteristics* section lays out the expected content, except for in-command editing. I would lump that feature as an important characteristic of modern command-line interfaces. (2a)
- This section could also have been supported by additional animations like the one you provide in *Component in Action*, except of course specific to tab completion or history. (2a)
- Thanks for saving space for *bash*—very much an obvious one for a future contributor to provide :) +(2a)
- Good choices for references, though localized citations would have helped show how you used them. Shneiderman's discussion of command interaction is the one key reference that I wish you used. (4d)

1a — + ...The implied binding to just operating systems is a miss, but not too big to be a spoiler.

1b — + ...Priority metrics, especially, are well done.

2a — | ...The entry is well-executed overall, though there are enough content gaps to mark improvement.

2b — + ...Kudos again to *Priority Metrics* here, but the overall discussion is spot-on and easy to follow.

4d — + ...Good sources here; Shneiderman and citing would have been good but are not deal breakers.

4e — You successfully issued a pull request. Commits are spaced and paced appropriately, with candid but sufficiently descriptive messages. (+)

4f — Submitted on time. (+)