Assignments: Principles 1

Charles Gulotta

cgulotta6@gatech.edu

1 QUESTION 1

The following subsections will discuss Piazza’s main “Q & A” interface. This interface was selected for its’ personal familiarity, utility, and high information density. Furthermore, our scope will be limited to a user interacting with the web interface using a mouse, keyboard, and monitor.

1.1 Processor Model

In general, the processor model views the user as a “sensory black box” which will produce output given some sensory input. More specifically, usability means the interface must be physically usable. Such a basic definition of usability means quantitative measurements can be recorded such as: “How much time does it take to complete a given task?” or “How many buttons interactable?”.

For Piazza, asking a question is a common task. This task can be completed by pressing 2 buttons. In this case, both buttons are presented in contrasting colors and can be clicked using the mouse. In my personal experience, this task is time bounded by the composition of the message, not the interface itself. The efficiency of this interface could be compared against other interfaces by requesting users to ask a specific question, thereby largely mitigating the time to compose the question.

1.2 Predictor Model

In contrast, the predictor model seeks to understand what the user predicts the outcomes of their actions will be and if the outcome of that action was what they predicted. Since this definition is more ambiguous than the processor model, the predictor model focuses qualitative measurements such as thought processes analyses.

Stepping back a level, a Piazza user may simply want an answer to their question. This means they can either find a satisfactory answer that’s already be posted or ask the question themselves. Personally, when searching for an answer, I use the page search function (hot-keyed to control + F) and enter a set of keywords. In my early interactions, I predicted these actions would search all the recent question and answer threads for matching keywords. This was not the case. Piazza instead provides a separate search bar to achieve this outcome.

1.3 Comparison

Using the processor model, I was able to understand the efficiency of using the Piazza Q & A interface to complete its' common tasks. Common tasks such as asking a question, or getting an answer, are quick to complete. This might lead to the conclusion that no improvements can be made to the interface in service to these tasks. However, from the predictor model, I observed that the interface did not lead me as a user to correctly predict the result of my actions. This suggests that the interface could funnel common search actions into the search bar.

2 QUESTION 2

The following subsections will relate to using Spotify to listen to music in multiple contexts.

2.1 Interactions in context

Listening to music is an activity that is pervasive through many common contexts across most cultural communities. A few of those contexts include:

- Waking up to a musical alarm
- Jamming out in the shower
- Enjoying the tunes at a dinner party
- Blasting your favorite workout songs at the gym
- Listening to a focus playlist while studying

In some scenarios, like waking up or falling asleep, the user may be unconscious and thereby unable to directly interface with the application. In other scenarios, such as after an intense workout or a shower, the user may attempt to interact with the interface while their hands or device are wet. This could translate to poor touch sensitivity or registration. Still other scenarios, such as group settings, involve multiple users with individual musical preferences.

2.2 Context Sensitive Design

In scenarios where the user is incapacitated, the interface could provide the option to automatically play and pause music according to a user defined schedule.

Alternatively, the application could leverage detectable parameters such as sunrise and sunset times or sleep-recognition data for automated control. Challenges posed by interacting with a wet interface could be addressed by adding support for non-touch-based actions, such as voice commands or gestures. In a group setting, the application could collect tracks user-suggested tracks from the local vicinity and play them on a nearby output device.

3 QUESTION 3

The following subsections will discuss the stages of the gulf of execution and the gulf of evaluation and how Canvas assists (or hinders) the user's ability to submit an assignment. For the purposes of this assignment, we'll assume the user is unfamiliar with Canvas specifically, but has some existing background knowledge based on experience using similar products.

3.1 Gulf of Execution

3.1.1 *Plan*

In this stage, the Canvas system should help the user translate their intentions (submitting their assignment) to actions inside the application. From the main dashboard, Canvas displays recent and upcoming assignments in a timeline view, color-coding them for each course. From this display, the user might grasp that courses and assignments are all separate. As such, the user would frame their intentions to upload their assignment to that page.

3.1.2 *Specify*

Given the assumptions gleamed from the planning stage, the user might plan to click on the specific assignment they wish to submit. The dashboard does not directly afford assignment submission to the student, nor does it signify that process. This leaves room for uncertainty at this stage. Since they've uploaded documents to other systems before, they might then plan to select their file from a file explorer window.

3.1.3 *Perform*

From this point, the Canvas system is responsive, clear in its signifiers, and offers numerous options for submission. These signifiers include contrasting color buttons, hyperlinks, and hover events (shading and bordering). Furthermore, the system provides feedback to the user when actions are prohibited (e.g. submitting a document in .docx format).

3.2 Gulf of Evaluation

3.2.1 *Perceive*

After clicking the “Submit Assignment” button, the user is prompted with a new page which says “Submitted!” along with the time of submission. Furthermore, the new page has replaced the original submission button with a “Re-submit Assignment” button. These signifiers are clear, concise, and are displayed without much other content which might distract the user. Additionally, hyperlinks are provided to view more “Submission Details” and to download the submitted assignment.

3.2.2 *Interpret*

Given the “Submitted!” message, along with the button changed to “Re-submit Assignment”, most users would conclude that the system has accepted their submission.

3.2.3 *Compare*

In most cases, the above interpretation would lead to the evaluation of goal completion. However, Canvas also provides numerous avenues to validate this interpretation by exposing the time of submission, “Submission details” and the ability to download the submitted file.

4 QUESTION 4

4.1 Large Gulf Example

Many popular role-playing games involve working with teammates to default a powerful enemy, or boss. In some cases, these bosses have abilities or attacks which can be thwarted or “interrupted” by the actions of a savvy player. In some instances, this is necessary to successfully defeat the boss.

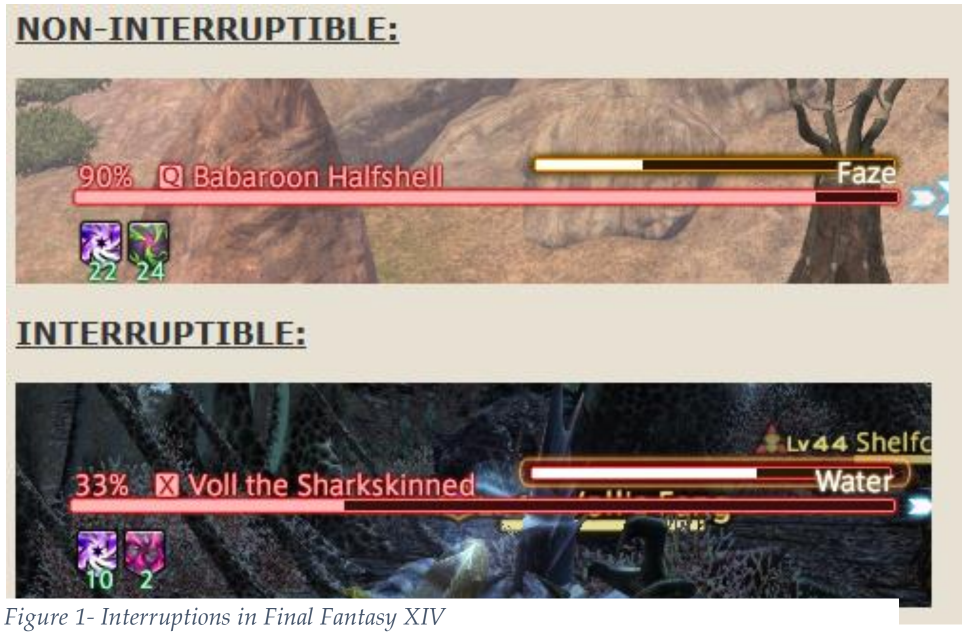
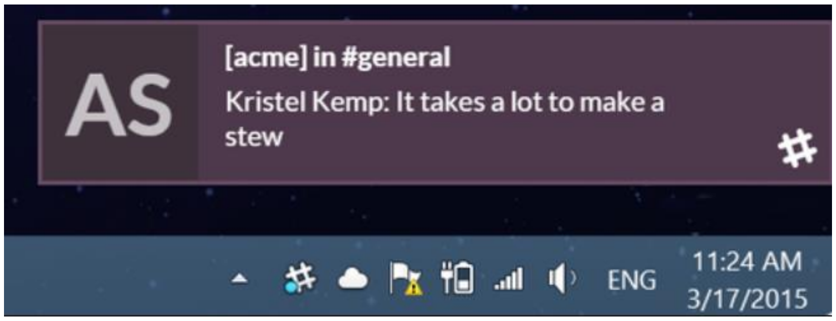


Figure 1- Interruptions in Final Fantasy XIV

Final Fantasy XIV offers this ability to its' players but doesn't adequately differentiate between actions that can be interrupted and those that can't. Figure 1 shows the uninterruptable ability "Faze" (top right) compared to the interruptible ability "Water" (middle right). The only signifiers are a slightly different color progress bar, and a lightly pulsing border. With numerous other HUD elements, particle effects, and lighting changes saturating the player's view, these signifiers are not sufficient to prompt a quick (< 2 second) response.

4.2 Effective Bridge Example

As part of my job, I am responsible for managing and assisting several teammates throughout the course of the day. Meanwhile, I still have my own work to focus on and accomplish. As such, responding to teammates is a frequent task for me.



AS
[acme] in #general
Kristel Kemp: It takes a lot to make a stew

Figure 2- Slack Notification Window

Since my team works remotely, we use Slack to communicate. Slack uses temporary fly-in windows to quickly notify the recipient that they have pending message. These notifications quickly allow them to assess:

- Who trying to communicate?
- What is the message?
- Do I need to respond immediately?

After a short time, the window disappears. This removes stale information from the users view, declutters the interface, and allows the user to return to their active task.

4.3 Application of lessons learned

In first case, the gulf of execution is large because the user is unable to quickly determine if their planned action (using their interrupt ability) will result in their desired outcome (stopping the boss' attack). In the second case, the interface effectively bridges the gulf of execution by allowing the user to plan their actions quickly. As such, Final Fantasy XIV could use a temporary, fly-in notification to prompt the user to interrupt the boss.