CS 6750: Assignment P1

Teng Xue
txue34@gatech.edu

1 QUESTION 1

As a Georgia Tech student, Canvas was selected to be evaluated in the following discussions as the most common interface we used in each semester.

1.1 Interface From the Processor Model of the User

From the processor point of view, humans are considered as a sensory processor, hence a good interface must fit with human limits. In terms of this, Canvas did a good job by organizing different functions/areas clearly to navigate users to proceed their task efficiently. For example, Canvas was designed to has a menu bar that included *Account, Dashboard, Courses, Groups, Calendar*, *etc*. This bar integrated most common tasks that users would do in the Canvas, putting them together would really benefit in quickly navigate users to their task. All the users need to do is to find the icon and click it.

1.2 Interface From the Predictor Model of the User

From the predictor point of view, humans are considered to predict what will happen as a result of certain actions they take. Hence a good interface must fit with human's knowledge and map input (expectation) to output (outcome). In terms of this, Canvas is a successful design to match user's expectation to outcome. For example, whenever I want to check grades of my assignments/exams, I expect to find them in "Grades" based on my experience. Additionally, I also want to check what the weight on each assignment/exam, I predict they should also be included in "Grades" based on my knowledge. And all these come out to be a complete match in the Canvas.

1.3 Comparison and Improvement

In the processor model, the design focused on how to accomplish different tasks more efficiently. Hence Canvas is suggested to be improved by displaying current enrolled courses in the main menu bar, rather than hide in the submenu bar. To be more specific, in the current design, users can only access the current enrolled course by clicking "Courses" or "Dashboard" first, then clicking twice

to enter a specific course portal to access *Home*, *Announcements*, *Assignments*, *Ed Discussion*, *Ed Lessons*, *Grades*, *etc*. Such design is less effective for users to accomplish their tasks.

In the predictor model, the design focused on what the users predict the outcome of their action will be based on their prior knowledge and experience. Such is very different than processor model, as the latter one only focused on website design to increase processing efficiency. Canvas is suggested to be improved by moving "Groups" under each specific course, rather than on the dashboard's menu bar. As a student, I would expect my group in a specific course should be in that course's navigation bar. While the current design might be more efficient, but it is not what users (students) predict.

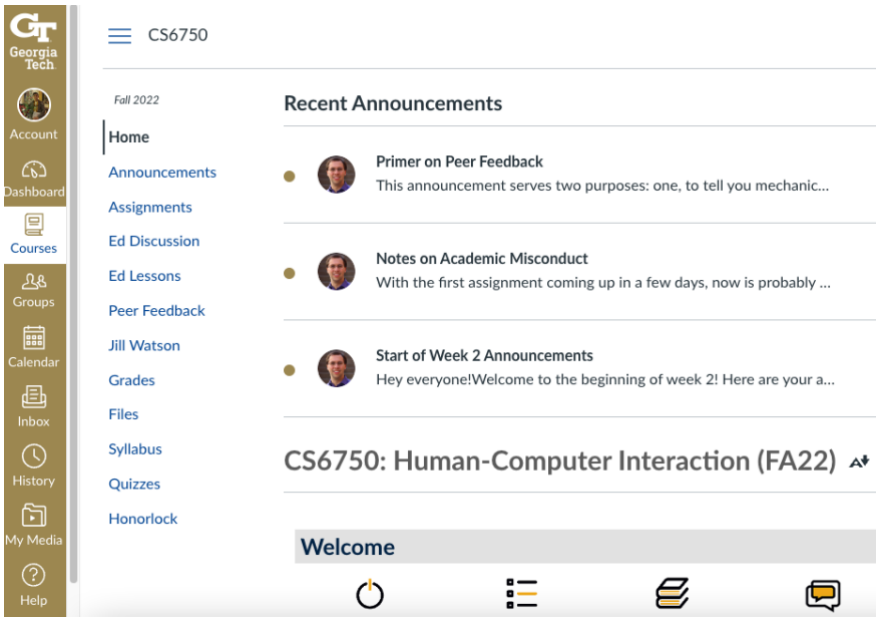

Figure 1 – Example of Canvas's Interface.

2 QUESTION 2

Music app was selected to be evaluated in the following discussions as the activity I performed with an interface in multiple contexts. For example, I always coding using my computer while listening to music on it at the same time.

2.1 Challenges to Use the App in Different Contexts

It's not difficult to image that different contexts surrounding the music app would add different challenges or constraints to use the app. In fact, various

contexts could distract my attention from the mu sic app interface to another work. For instance, I maybe coding with listening to the music, then I want to focus on coding while I can still see lyrics without switch between different interfaces. I maybe driving with listening to the music, then I want to pick and play my favorite music while both of my hands are probably occupied. I maybe running with listening to the music, then I'm likely to use earbuds or AirPods to control the music playing without looking at the interface in the app.

2.2 Improvement on the Interface Design

In order to make sure the music app perform differently on various context to overcome certain challenges or constraints, the design of the interface is important. First of all, in the coding scenario, the interface should be minimized as a control bar with lyrics displaying on it. Such design allows users to control the music app without switching between different interfaces. The control bar interface could be moved to anywhere on the desktop that does not interfere with users work but could still allow users seeing the lyrics if they want. Secondly, in the driving scenario, the interface should be designed as voice controlled. In this way, users could give order to the app verbally to play any songs they like or increase/decrease the volume of the music according to different surroundings. Lastly, in the running scenario, the interface should be designed as a communicator in the app with earphones. Taking AirPods 2 as the example, AirPods 2 was designed to allow users controlling music on the surface of AirPods, like tapping the surface twice would switch a different song. Hence the music app interface must be designed to follow the order of AirPods, under the Bluetooth connections, to let users control the app without the need to use the device app.

3 QUESTION 3

3.1 Gulf of Execution

3.1.1 *Identify Intentions*

When student submits a question to the Ed Discussion, the first step is to identify the intentions in context of the Ed Discussion: submit a question through the Ed Discussion website. There is very straightforward, no mismatch exists between student's understanding and the interface of Ed Discussion.

3.1.2 *Identify Actions*

Student then need to identify the actions necessary to accomplish their goal. First, student could click "New Thread" button on the top corner to start a new post. Second, student would follow instructions to post a new question: Fill out *title*, choose a related *category*, edit the main *paragraph* of the question, choose an option to post in *private* or *anonymous*. Ed Discussion interface does a good work to help student understanding the actions necessary to realize their goals.

3.1.3 *Execute in Interface*

The last step for the student to do is to execute within the interface. Once the above steps had been executed, student could hit the "Post" button on the bottom corner to submit this post. This step is also straightforward, no difficulty exists in Ed Discussion interface. Overall, Ed Discussion successfully carries the student across all stages and the gulf of execution is short.

3.2 Gulf of Evaluation

3.2.1 *Interface Output*

Once a question was submitted, the first thing student want to do is to check the interface output. A new post would be displayed on the interface of Ed Discussion immediately after student hit the "Post" button. This new post shows the numbing (#) of the post, title, time, as well as different options: *Comment, Edit, Delete*. Hence there is no challenge for student to understand the interface output from Ed Discussion.

3.2.2 *Interpretation*

With the interface output in hand, student then need to interpret these. However, I think Ed Discussion fails to communicate the outcome of the actions. Since there is no clear check mark that helps the student to interpret if their submission succeeds or not. The only way to interpret is assume it succeeds as no error message was shown, and check if the number of the post is the greatest (means the latest post).

3.2.3 *Evaluation*

The last step for the student to do is to evaluate if their actions accomplish the goal. Ed Discussion shows "VIEWS" for each post which could help student to

evaluate that. Since the number of views would increase with time passed by if the post was successfully submitted. Or the question could be answered by other students or TA/Instructor which is also a valid evaluation.

4 QUESTION 4

4.1 Activity with Large Gulf of Execution

When I am driving, an activity with large gulf of execution is trying to solve the tire pressure warning on the interface of the dashboard. Gulf of execution includes three stages: identify intentions, identify actions, and execute in interface. There is no difficulty for me to identify intentions, both my goal and the interface warning is to fix tire pressure issue. However, the wide gulf comes from identify actions and execute in interface. Except from flashing the warning on the dashboard interface, there are no instructions on identifying actions. I have no idea which tire has a pressure issue, how bad is that, and how to fix that. As a result, I have no clue to execute in the interface.

Figure 2 – Example of Tire Pressure Warning with Large Gulf of Execution.
Source: https://autovfix.com/car-tire-pressure-light-still-on-after-filling-tires/

4.2 Activity with Small Gulf of Execution

Still taking driving as an example, an activity with small gulf of execution is to solve the seat belt warning on the interface of the dashboard. Analyzed by three stages in the gulf of execution: identify intentions, identify actions, and execute in interface. The intention is straightforward: buckled up to be safe. The following action are easy to be identified: First identify which seat belt is not buckled up (driver or first passenger seat) Then buckled up. Execute in interface is not necessary as the warning would disappear once buckled up.

4.3 Lessons Learned to Resolve the First Wide Gulf

To resolve the wide gulf in the first activity, some lessons could be borrowed from the second activity's interface. First of all, once the tire pressure warning on the interface of the dashboard, clearly identify actions that users need to do. For example, which tire has a pressure issue (left front, right front, left rear, right rear). What's the pressure reading of that tire and how long it can keep drivable. Then instruct the user the correct way to handle in different pressure readings. For instance, stop immediately when the reading below 30. Or slow down if the reading below 35. Finally, navigate the user to execute in the interface on the dashboard to reset the warning.