Assignment 1
CS6750, Fall 2020

Prashant Shinde
pshinde32@gatech.edu

***Abstract*—**Assignment P1 explores students understanding of "Design Principles" and "Feedback Cycle" concepts of Human Computer Interaction course. Following paper will answer 4 specific questions asked in assignment description.

QUESTION 1:

As a Georgia Tech OMSCS student, you likely regularly use Canvas, Piazza, Udacity, and other tools. Select one of these interfaces (or one piece of one interface).

First, discuss that interface from the perspective of the processor model of the user. The processor model is concerned with objective, measurable outcomes, so note the efficiency with which you can accomplish different tasks.

Second, discuss that interface from the perspective of the predictor model of the user. The predictor model is concerned with what you predict will be the outcome of your action, and whether you can interpret whether the outcome matched your prediction, so focus on how the user perceives and interprets what they should do and whether it was successful.

Finally, compare the insights you gained from each of the previous discussions. What sorts of improvements would the processor model suggest, and how do they differ from the sorts of improvements that the predictor model would suggest?

Q1 Answer:

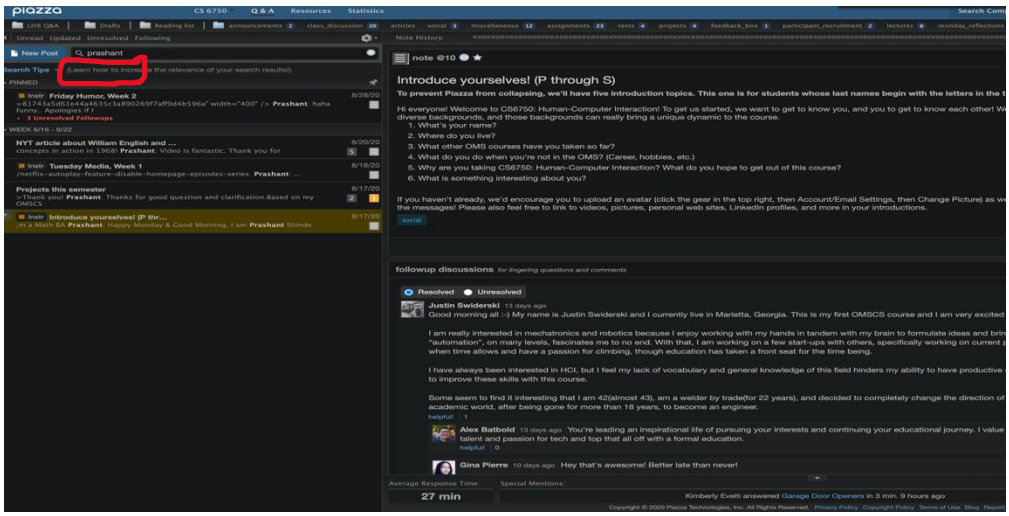
To answer this question, I choose 'message threading and cascading' feature on Piazza.

Processor Model: Assume I am trying to search for a post/discussion related to GitHub (It's titled as 'Guide to getting started with Overleaf, JD & Git' on Piazza).

Using Piazza User Interface, user can 'search' the word using 'search bar' provided.

It is extremely intuitive and highly efficient to sort through & search for a post in a corpus of thousands of interactions between students in class.

BUT in the other hand, if I want to search for a specific post by me using my name 'Prashant Shinde' in search bar then Piazza returns the post in which your name appears but not 'the comment by you'.



In above, Piazza returned all posts done by a user, but it fails to provide exact comments done by user. A user was expecting their 'exact comments' & not 'a post containing their comment'. There could be 1000 comments on a single post and your comment maybe let's say at rank 600, then you have to scroll all 599 comments to reach your own comment. This is against the 'expectations' vs 'results.

Predictor Model: Action of 'searching an old post using keyword' should result in 'expected post' itself. As explained above in example 2, name search didn't result in expected outcome. In case 1, processor model & predictor model bridges & narrows gulf of execution & evaluation but in case 2, it fails to do the same. A predictor model suggest outcome as 'actual post by user' and then link it to main thread.

Based on both processor & predictor model, special cases like case 2 can be redesigned with user's results intention in mind and narrowing the gap of evaluation by considering all use cases.

QUESTION 2:

When we take the participant view of the user, we look at the entire context surrounding their activity. However, some interfaces are designed for activities that exist in different contexts.

First, select one activity you perform with a computer interface in multiple contexts. For example, you might use a text messaging app while walking down the street, sitting at your desk, eating a meal, or driving your car (though hopefully not!). You shouldn't select text messaging: select some activity that is not used as an example in this question prompt.

Then, discuss how the different contexts surrounding the app add different constraints or challenges to using the app. For example, with a text messaging app, your cognitive resources are more divided eating or driving, and your physical precision is less reliable while walking.

Finally, describe how the design of the interface might be altered to perform differently depending on your context to overcome those constraints. You may assume that the interface is able to magically know your context. For example, if you were writing about text messaging, you could assume the app would know if you were driving, walking, or sitting.

Q2 Answer:

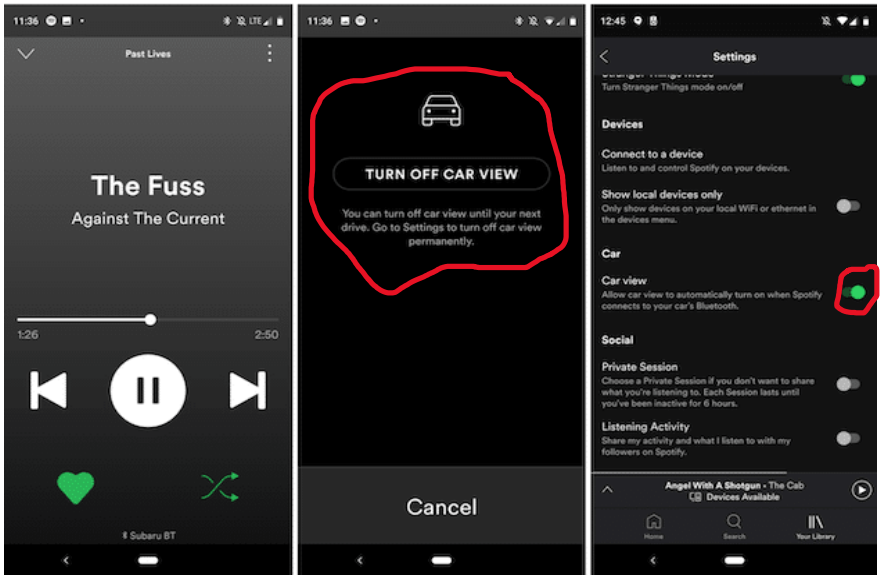
To explain a UI design constraint in multiple context of my daily activity, I choose User Interface of Spotify.

When a user is driving a car and phone is connected through a Bluetooth, Spotify UI changes to 'Drive Mode' and limits all its functionality until your phone is connected to car speakers through Bluetooth. All phone calls will be disabled, and user cannot access playlists or other music by interacting with Spotify UI. The only functionality user can do is 'play next or previous song'.

When used normally (not inside car), you can search for song, artist, explore new playlists, etc. So, on aggregate you can think as 2 Spotify UIs:

1. Normal Mode (Full functionality)
2. Driving Mode: Car View (Limited functionality: Frozen UI)

In driving mode, the app automatically detects a Bluetooth connection with vehicle and alters the UI.



I can see 2 potential re-design solution to UI:

1. Based on user's previous app-drive usage history/pattern, app should automatically know and just disable 'Car View' for specific users
2. To disable the 'Car View' feature of Spotify, a user navigation within settings can be reduced and a simple 'disable' option can be intergraded on UI when a 'Car View is enabled automatically

Spotify's intentions of users are positive, as they expect users to interact less with app while in Car, but it totally makes a UI journey cumbersome by increased navigation to disable the 'Car View' option.

QUESTION 3:

Describe the process of submitting an assignment to Canvas in terms of our discussion of feedback cycles.

For each of the three stages of the gulf of execution, describe how Canvas either successfully carries the student across that stage, or in what way it fails to carry the student across that stage. Assume the student does not already know exactly what to do: how does the system help them figure out what to do?

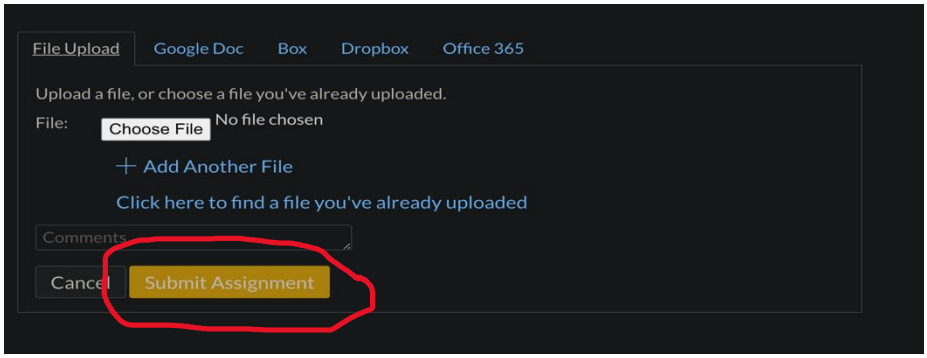
For each of the three stages of the gulf of evaluation, describe how Canvas successfully communicates the student the outcome of their action, or in what way it fails to communicate the outcome of their action. Again, assume that the student does not already know exactly what to expect.

Q3 Answer:

3 stages of gulf of execution are: Plan, Specify, Perform

Canvas has a simple User Interface which is straight forward to use. I have been using Canvas for last 2 semester hence I might have developed a cognitive bias towards simple, rich Canvas UI. But I think Canvas does a great job of solving a task simply, which is 'Assignment Submission'.

The Goal: Canvas UI is simple straightforward and 'Submit Assignment' button is visible clearly to execute the task



Plan: Canvas let student choose file upload options from Google, Box, Dropbox, Office 365 or from local machine. In terms of assignment submission, a student can choose one of the options

Specify: Canvas let student recognize what actions they can do easily

Perform: After file upload, user can click on 'Submit Assignment' button to complete the task

One challenge that I have observed with Canvas is the colorblind effect of 'Submit Assignment' tab.

The Yellow color is not suitable for colorblind people and hence might add some challenge while trying to complete a task.

3 stages of gulf of evaluation are: Perceive, Interpret, Compare

Once the assignment is submitted, Canvas clearly notifies student on the UI about date & time of submission and submission status. This feedback loop gives user an assurance of their assignment submission.

Also, you will get an email through Canvas (if option enabled) once the assignment is submitted successfully.

QUESTION 4:

Select an activity from your regular life that struggles with a large gulf of execution or gulf of evaluation, especially due to a weakness of the interface involved in the activity. First, describe what makes that gulf wide. What are the failures of the current interface to bridge the gulf?

Then, select a different, but similar, activity from your regular life that does a better job bridging its gulf of execution or gulf of evaluation. Briefly describe that activity and what gives it a narrower gulf.

Finally, describe what lessons could be borrowed from the second activity's interface to resolve the wide gulf in the first activity.

Q4 Answer:

2 light lamps in my apartment are controlled using Google ('Hey Google'!). Lamp A is a simple mono control (ON/OFF) lamp and lamp B is multi-control (brightness adjustability).

Every time I say, 'Hey Google, Turn on Lamp A', it just works perfectly fine, but Lamp B is always a struggle due to *gulf of execution* issue. Both the lamps are connected to a WiFi switch which is eventually connected to Google Home.

Due to multi control functionality, Lamp B is NOT always on OFF mode and hence when I say, 'Turn On', the WiFi switch gets ON but not lamp B. Lamp B has *ON, brightness level 1, brightness level 2, brightness level 3* & *OFF states*.

On the other hand, lamp A has only 2 states, ON & OFF. Hence, every time I say 'Turn On' it is either *OFF or ON,* hence it works just fine.

To summarize, it's not a Google Home or WiFi socket issue, it's the lamp engineering design issue.

Lamp B narrows this gap on expenses of sacrificing multi-mode brightness functionality.

I think, manufacturers of Lamp B should keep their existing multi brightness

Lamp A

Lamp B

control functionality but when a lamp is turned OFF (here WiFi switch to lamp is connected), lamp should RESET to level OFF (like lamp A), which is not the case currently.

It will be a simple fix in design engineering & it will save me a lot of time and will give mental satisfaction.

To summarize:

Task: Turn ON light lamp

Issue: Gulf of execution in Lamp B due to engineering design.