Assignment P1 (Summer 2020)

Deepti Venkatesh
dvenkatesh7@gatech.edu

1 QUESTION 1

The interface I choose for answering the first question is *Canvas*.

1.1 Processor Model

For the processor model, we can take into consideration the *Quizzes* section in Canvas. The student starts answering the questions in the quiz and the result is displayed as soon as he clicks the Submit button. The student can immediately know his performance and take the test again to improve his score.

Another example can be that of a survey. The user is presented with survey and he fills it out and submits it.

1.2 Predictor Model

For the predictor model, we can take into consideration the *Modules* section in Canvas. After the student is done watching the current video, he can click on the Next button to move to the next video. When he hovers on the Next button, it says what is the content in the next page. If it's a normal video of next topic in the lesson, the student can predict that he will be taken into another video and the student can continue to take notes but if it is a Quiz, the student can predict that a quiz is coming up and he can get prepared to answer the quiz.

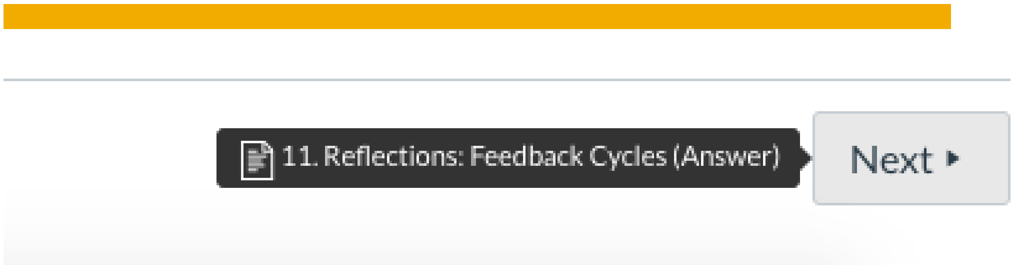
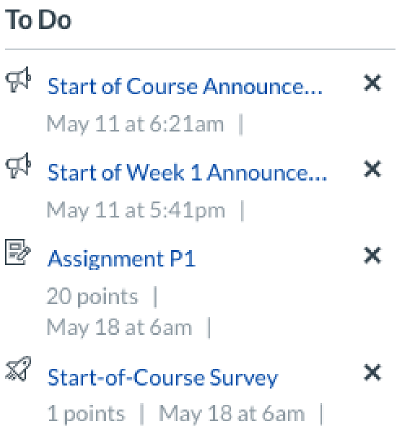


Figure 1—Figure shows the Next button in Modules section of Canvas.

For the predictor model, we can also take into consideration the *To Do* section.

It lists the tasks that the student needs to complete according to their due dates. So the student knows that they will be taken to the task that they select from the list.



To Do

Start of Course Announce... ✕
May 11 at 6:21am |

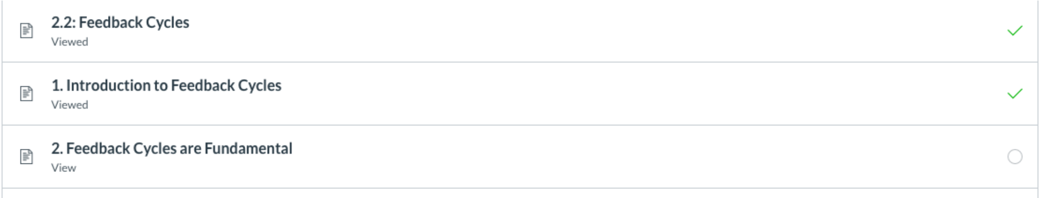
Start of Week 1 Announce... ✕
May 11 at 5:41pm |

Assignment P1 ✕
20 points |
May 18 at 6am |

Start-of-Course Survey ✕
1 points | May 18 at 6am |

Figure 2—Figure shows the To Do of Canvas.

We can also take into consideration the "Modules" section in Canvas. This section has a list of all videos under each module and next to each video is a small space which either contains a green tick mark which indicates the student has watched the video or a blank which indicates the student has not watched the video yet. So when the user sees that he can clearly figure out what videos he has watched and what videos he is yet to watch. He can also be sure that after he has watched a video, a green tick mark will be displayed.



2.2: Feedback Cycles ✓
Viewed

1. Introduction to Feedback Cycles ✓
Viewed

2. Feedback Cycles are Fundamental ○
View

Figure 3—The figure shows the green tick marks and blank which help student understand where he left off.

1.3 Comparing Insights

If we consider the quiz or survey in processor model, we can get insights like how much time on average a student takes to complete the quiz, the mean and median score of the class.We can also gather details on how many users took the survey in the class.

If we consider the predictor model, if we know that the user is not clicking on the "Next" when the hovering text shows "Quiz", we can gather insight that the user might want to take the quiz at the end of the lesson rather than in between the lesson. So from next time, the app can improve by structuring the lesson to put the video content at the beginning and the quiz at the end.

From the above discussions, we can say that the insights gained from the processor model is more quantitative. We just get the statistical data and not the reason behind that data. So this cannot be used to make improvements to the design. Whereas, using the insights that we gained from predictor model, we can make changes to the interface to better serve the user.

2 QUESTION 2

The activity that I will discuss is the Video Player like YouTube.

A free video streaming app like YouTube is going to show a lot of ads while playing a video. Even though the *Skip Video* option is available, often times the user will not be able to pick up the phone and click on it to skip the ad and continue playing the video. Below are some examples of such contexts.

2.1 Constraint 1

Constraint - While the user is exercising or running, he would play a video and keep the phone away in his pocket while running or somewhere close to him in the gym.

Solution - When the Skip Video option is available, the user should just be able to say *Skip Ad* and the app should recognize it and skip the ad and start playing the video.

Design Alteration - The interface should incorporate the audio command feature.

2.2 Constraint 2

Constraint - While the user wants to listen to music while cooking, they will keep the phone away from the cooking area and because of the added noise of the kitchen vent, microwave, the app might not be able to recognize the *Skip Ad* command from the user.

Solution - Under such circumstances, the app should be able to skip the ad without the interference of the user.

Design Alteration - To achieve this, YouTube might ask the user if he wants to skip ad after 5 seconds by default as soon as he opens the app.

2.3 Constraint 3

Constraint - The other context is while putting a baby to sleep. Often times I use YouTube to play *White Noise* to put my baby to sleep. When a loud ad plays amidst the white noise it is just going to disturb the baby and eventually wake her up.

Solution - Under such situations, the the volume in which the ad plays should be reduced completely and once the ad is skipped or played completely, the volume should be brought up to its original level to continue playing the white noise.

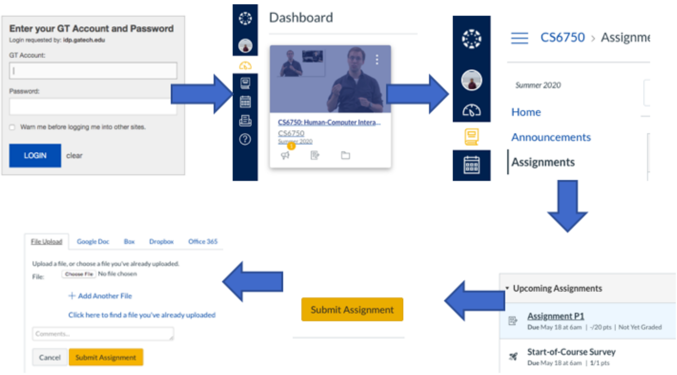
Design Alteration - To achieve this, the app should be able to ask the user if he wants to mute the ads and if the user chooses Yes, the app should reduce the volume when the ad plays.

3 QUESTION 3

3.1 Gulf of Execution for submitting an assignment on Canvas.

First the user has to identify the intention in the context of the system. In this case it is submitting the assignment on Canvas.

Second, the user has to identify the actions that he needs to perform. Canvas is nicely designed and hence it is easy for the user to know where to start. The user first logs in to Canvas and clicks on the HCI card on the dashboard. Since he wants to submit an assignment, the first thing he will be tempted to do is click on *Assignments* menu in Canvas. On the next page he sees the list of assignments available. The user knows which assignment he is trying to submit. Hence, he will click on that assignment. On the next page, there is a big orange button *Submit Assignment* which catches the eye of the user. So now he knows he has to click that to submit the assignment. Once the user clicks on it, options to upload the file is provided. This is one place the user has to spend some time to figure what to do next because, the file upload option is somewhere at the bottom and it is not high-lighted like the *Submit Assignment*. But once the user figures out that step, finding out next steps is easy. If the user selects a file type that is not supported, a small alert is displayed to inform the user about the document formats supported. Even if the user is not quite used to using Canvas, it is very clear to him that he needs to browse for the correct file which he needs to submit, upload it and click on the submit button. Now the user has understood what steps he needs to follow to submit the assignment. The steps are as shown in figure below.



Upload a file, or choose a file you've already uploaded.

Figure 4—The figure shows the steps identified by user to submit an assignment.

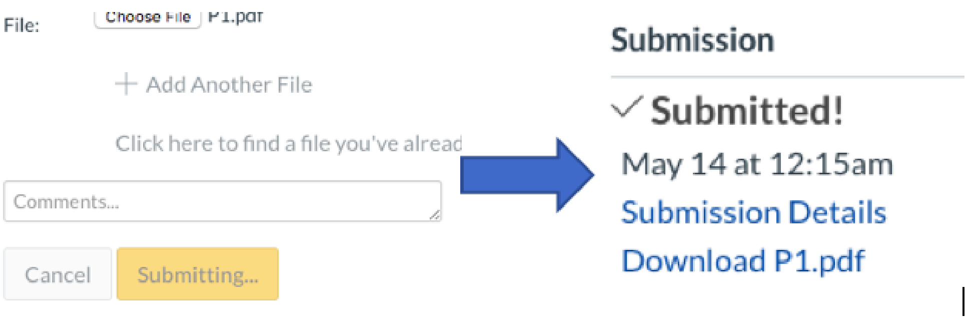
The final step is executing all the above identified steps. All these steps can be performed without any prior experience of Canvas usage. Hence this can be considered to be a very good design.

3.2 Gulf of Evaluation for submitting an assignment on Canvas.

Once the user submits the assignment by clicking on the "Submit" button, the Canvas interface first displays *Submitting* and upon successful submission, the *Submitted* message along with the date and time of submission is displayed as confirmation.

Second, the user can easily interpret this result as - the assignment has been submitted successfully on the displayed time and date.

Lastly, to evaluate if the desired action has indeed been performed successfully, the user can log out from Canvas and re login and come back to the Assignment section and click on the assignment that he has submitted. A message *Submitted* will be displayed next to the assignment along with date and time which ensures the user that the assignment has indeed been submitted.



Submission

✓ Submitted!

May 14 at 12:15am

Submission Details

Download P1.pdf

Figure 5—The figure shows the message that appears on successful submission of an assignment.

From the above explanations, it is very clear that even though the user has no knowledge on how to navigate in the interface, the interface itself is quite intuitive in guiding the user what to do.

4 QUESTION 4

4.1 Broad Gulf

For the example for a device with broad gulf, I would like to talk about the fan/light in my dining room.

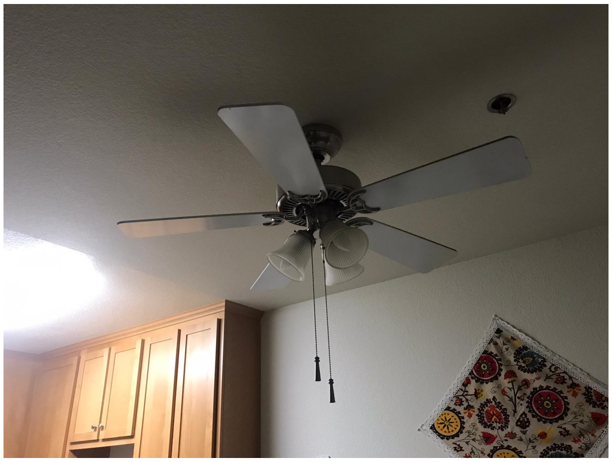


Figure 6—The figure shows the fan/light.

This is supposed to be fan/light and it has two strings which we have to pull to operate it. When the user sees this for the first time, he cannot make out which string controls the fan and which string controls the light. He has to pull both to figure that out.

When we talk about this device with respect to gulf of execution, the user's intent is to switch on the light. But he is not able to identify the steps to accomplish his task as he does not know which string controls the light. This creates a broad gulf between the intent and the execution.

The other problem with this design is controlling the speed of the fan. If you pull the fan string once, fan runs in speed 3, if you pull it again, it reduces to 2 and then to 1 and then the fan stops. How is the user supposed to know this without someone telling him? And if we want to change the speed from 2 directly to 3, we have to first pull the string to change it to 1 and then pull to stop and then pull again to set the speed to 3. Also, there is no way to figure out the current speed in which the fan is running. The design is not intuitive at all.

The above reasons creates a huge gulf between the intent and execution as the user cannot identify the steps.

4.2 Narrow Gulf

For the example for a device with narrow gulf, I would like to talk about the electric pressure cooker, Instant Pot.

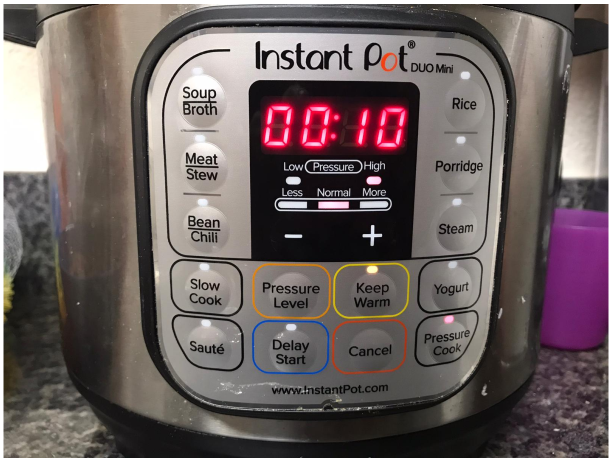


Figure 7—The figure shows the controls of Instant Pot.

As we can see in the figure, user can easily identify which mode is on from the red light. So if the user's intent is to pressure cook, he knows that he has press the *Pressure Cook* button. There is + and - buttons to set the cooking time. The display shows the time that is set and the time remaining for the cooking. User can easily increase or decrease the time using the + and - buttons.

When we talk about this device with respect to gulf of execution, the user's intent is to pressure cook, the step is easily identifiable because of the naming. Setting time is also very intuitive because of the presence of the + and - buttons and hence there is no gulf between the intent and execution.

4.3 Bridging the gulf

To reduce the gulf in the first example, we can borrow the idea of labeling and providing controls from the second example. The system can have some sort of display for displaying the fan's current speed and knobs to increase and decrease the speed. Also, the string can be labeled as FAN or LIGHT to make it easy for the user.