Assignment P3

Gaurav Pande

gpande@gatech.edu

Question 1

Invisible Interfaces

Flexibility- Do whatever feels flexible/comfortable to you

Flexibility offers users to perform operations on the task in a manner in which a user is comfortable with their experience on the interface. Flexibility takes into account the type of experience a user holds and offers them options in which they feel comfortable and fast. By not forcing a set of operations to a user, flexibility allows a user to identify and perform operations in more than single way or in some other way for which the user has already seen in some other interface and comfortable with. For example, the Undo/Redo operations now have the same symbol and keyboard operations in most of every platform. Like shown in figure 1, novices can use the icons to undo/redo operations and an expert may want to use keyboard shortcut ctrl+z/ctrl+y hotkeys, and these operations are same in many other interfaces like google docs, Microsoft word, adobe photoshop e.t.c.

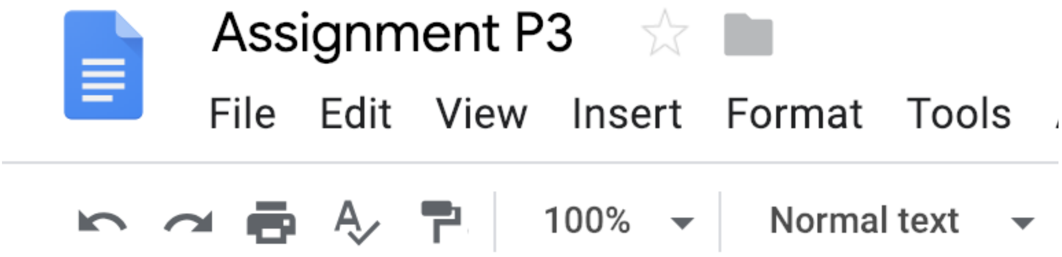


Figure 1: Icons shown in the leftmost for Undo/Redo operation in google docs.

Consistency- Indulges you in a habit of using things easily

One of the most important characteristic while designing invisible interfaces is I think is consistency. The user does not need to learn new things for the interfaces in which the designs are consistent to do common tasks. This help in reducing the Gulf of execution because you do not need to find the operations or task in the interfaces as the user already know how to perform the task.By relying on the consistent knowledge the user is not searching through the interface and does his task in a seamless manner without any interruption of finding and searching.

Discoverability- Easier to find things make execution fast

Discoverability offers the user a sense of finding things easily even if they come across the things for the first time because the design considers the pre-knowledge of the user experience and prototype solutions where the tasks are easier to find. It also reduces the gulf of execution significantly as the intuitive design takes away the cognitive load and time to search through the interface to find operation, thereby making interface invisible in practice.

Participant View

With respect to the participant view, one of the most important design consideration is the **comfort** of the user while using the interface. It not only reduces the cognitive load from the user but also make sure that regardless of the abilities of the user it is easy to execute or perform. The best example would be the **"Reachability"** feature in iPhones using which a short swipe down or a quick swipe up and down near the bottom of the display brings the top half of the screen down so you can get to it more easily with one hand. It doesn't require you to take your thumb throughout the screen rather it restricts your phone layout in a smaller screen where you feel comfortable to execute.

Another design principle of **Tolerance** tells a participant view to consider the distractions that a user might face while using an interface. Distractions make a user prone to make errors or mistake, and if an error occurs Tolerance should handle those mistakes and revert back the interface to a safe state. For example, most of the modern applications like SoundCloud, Netflix, if a user hit the close button unexpectedly, these applications save the state of the movie/song you were watching. Next time you open the application you can resume the song or movie from where it was closed last time due to the distraction.

Question 2

Command Prompt in MacOs: Intolerant to errors

I guess most of the command prompt whether it is Linux/Windows/Mac(I am choosing Mac cmd for this specific question) are all prone to errors because of their Unix shell style design. Being a computer science student a lot of time it happens:

1. That you forget in which directory level you are and you accidentally delete a file using “_rm -rf filename_” command, the file gets permanently deleted and unrecoverable, and if by any chance you are in root and did “_rm -rf*.*_”, then dear lord!! you have just wiped everything from your system.
2. That if you are a coder and familiar with git[1](a tool to upload your code to a cloud repository), then you must have some time trouble pushing code initially from cmd. If you do not know which branch you are, and you do just “_git push_” you may have pushed your development branch to the master branch. God that is a blunder!
3. What if you made an error and you do not know you have made an error. This can happen all the time while working on a command prompt. You hit “_git push”_, but the code didn't push due to many complex factors.

Avoiding errors: Mapping

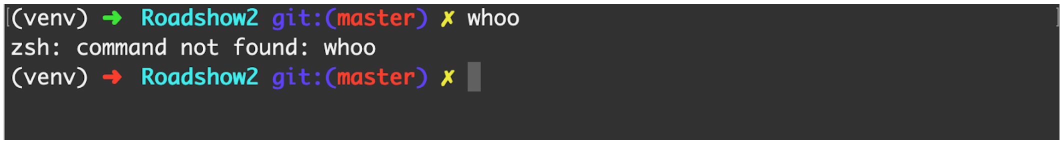
Mapping efforts to make the task mapped to their intended results, like for example in our above case we can just make the command line tool with a different command to permanently delete an item using the renamed operation of “_rm_” to “_prm_” or “_rm -p_” in which the user at least can easily distinguish that firing an “_rm -p_” means a permanent delete while doing an “_rm_” means temporary delete.

Avoiding errors: Affordances

Introducing Affordance with the help of signifiers in your design can help you to identify visually that some error happened while you typed the command. It can help reduce not only cognitive load but also helps in making the interface invisible as the user know quickly where he is and what happened. In our example what we can

[1] https://git-scm.com/book/en/v2/Getting-Started-The-Command-Line

do is; we can show a small arrow-like "->" signifier in green color whenever the command is successfully executed and in red color whenever an error occurred. In addition to this, we can showcase the name of the branch we are when we browse through the git directory as shown in figure 2. This is the replacement tool known as iTerm[2](a customized tool with different themes)



```
(venv) ➜  Roadshow2 git:(master) ✗ whoo
zsh: command not found: whoo
(venv) ➜  Roadshow2 git:(master) ✗ ▮
```

Figure 2: iTerm utility showing colored indication of the command you execute. Red means you fired a wrong command. Also, observe the branch name in the bracket.

Avoiding errors: Constraints

What if you can reduce the errors while you type a command. Constraints in the design can help us in reducing the errors and restricting the user to execute the operation in a safe-fail manner. In our case what if we have a feature like "google autocorrect" where when you type a command it autocorrects you and let you type the command faster(which will reduce the Gulf of execution) and prevent you from typing an incorrect command. What if, when you type the name of a directory it lets you restrict you and suggest you automatically the directories starting with that name in the terminal. Wouldn't that be nice to as you won't have to do "*ls*" to list the directories!

Question 3: slips, mistakes, and errors

Introducing "2048"

2048 is a very famous mathematical game where one has to reach score 2048 by merging of different tiles appearing on the screen as seen in figure 3. The target in this game is to score 2048, by moving the tiles LEFT, RIGHT, UP, DOWN using the touch screen of the phone. Whenever you move 2 similar tiles they both get

[2] https://www.iterm2.com/version3.html

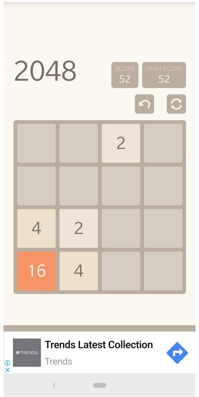
merged into a single tile with the score of summation of 2 merged tiles. So the game is simple: move the similar together and try to score 2048.

A slip

In this game when you move 2 similar tiles, their numbers get added and they get merged. So a user knows from a mental model that where he or she needs to move the tile in order to get the maximum score. However, the user may still slide in the incorrect direction. This is due to the action **based slip.**

The reason for this error to happen can be due to the lack of experience of the user or due to the sensitivity of the touch.

To correct the slip, the user needs to gain experience or calibrate the touch sensitivity.



		2	
4	2		
16	4		

Figure 3: 2048 game interface

A mistake

This game is also prone to **rule based mistake,** where the user is not aware of the mental model and make mistakes. Every turn, a new tile will randomly appear at the random position on the board with a value of 2 or 4. So a user is not aware of where the tile might appear on the screen and makes a mistake initially if he does not know about this rule. So they make the **wrong slide** based on the **wrong mental model** and then make their subsequent slides based on the new tile they get.

The reason for this mistake I think is due to the inexperience of the user. Novice user is more prone to this error.

To correct this error, one needs to play the more often to increase the experience and enhance the mental model.

Other Challenges

The grid started to fill as soon as you start playing and soon you are left with few open positions to slide, which make it infeasible for movements and also the user get confused now as they do not know in which direction to move as some of the tiles can get merged but due to lack of space the mental model gets disturbed.

Question 4

Good representation

"**Trello**" is an application for project management or personal task handler for your day to day tasks. This is like a kanban board for better visualization and management of your task to increase your project efficiency. The main advantage of this application which makes it good are:

Explicit relationship/mappings: You can add task by customizing cards as seen in figure 4. You can simply add the task for "TODO", ”DOING" or "DONE" and simply drag and drop them. You can color code your task with its priority and assign those task to the team members. The mapping of the task explicitly makes this application easy to use

Exclusion of extraneous elements: You can simply observe in this application that all the complications and extra buttons and features which are there in traditional kanban boards like jira[3] are hidden and abstracted to give the user comfort to use this application in the simplest manner. You do not see any bug or subtask or statistics or formulae to add on the screen. You can simply delete your

[3] https://www.atlassian.com/software/jira

task by archiving the task which makes this board a great use for managing your personal projects and day to day tasks.

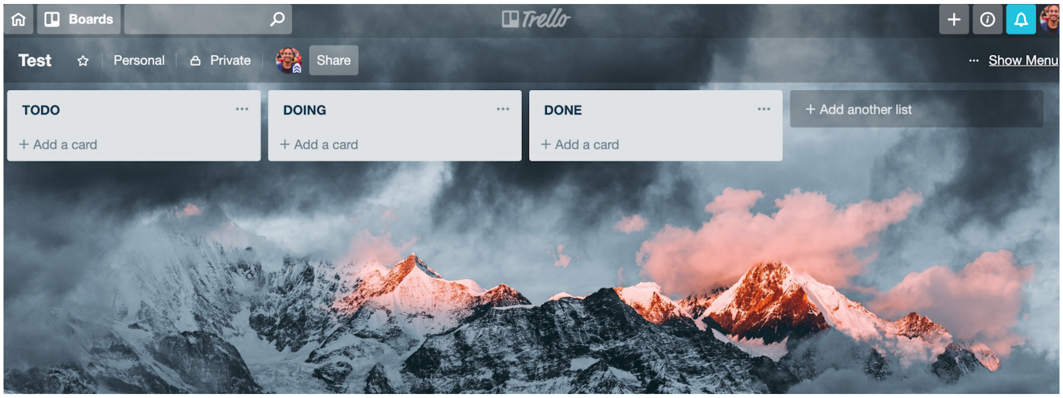


Figure 4: A typical trello board with the task as “TODO”,”DOING”,”DONE”

Bad representation

One of the interfaces that I feel like lacking the good representation is in the traditional MacBook keyboards buttons for volume and brightness control. As you can observe in figure 5, the design lacks the following characteristics:

Explicit mapping: The design does not have an explicit mapping to indicate the level of the volume of brightness changes. You can keep on pressing the increase volume key but there is no mapping which could tell you what level of loudness you have reached.

Constraints: The interface does not offer the scaled representation of the action performed. There is no limit set where you get an indication saying this the maximum now you can stop pressing a key. Like in the mobile phones when you select a ringtone, there is an indication using a sound played to indicate the loudness you have selected, similarly, there can be a design where we can showcase in the MacBook keyboard as well to denote the levels of loudness.

You can observe that in figure 6, which the interface of the new MacBook touch bar offers a good representation of the indication and constraints to the user by sliding the finger on the touch-based volume control.

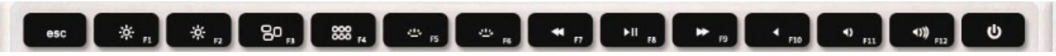


Figure 5: Traditional MacBook keyboard with volume and brightness control

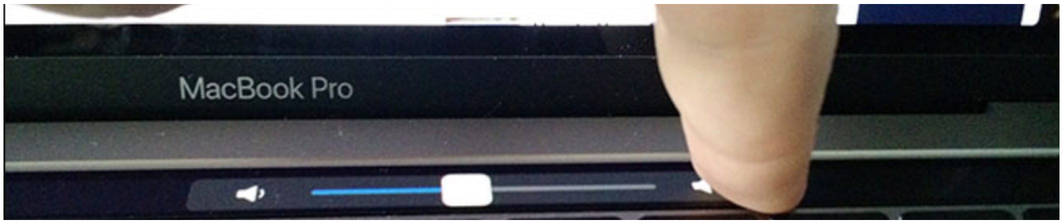


Figure 6: The new MacBook touch bar with scaled volume and brightness control