Assignment P3(Fall 2018)

Chamikara Dharmasena
cdharmasena3@gatech.edu

Question 1

Simplicity

Simplicity would be one of the principles that would help to narrow the gulf of execution and evaluation. If the design is easier to understand and a user is only given the information they need, gulf of execution will be much smaller. If the result of the user's action is presented in a simpler manner, gulf of evaluation can be minimal. Simplicity in design isn't just about the minimal colors, buttons or whitespace, it is about understanding the user's mind and design a product that rids itself of inconsequential elements and widens the gap between the user's goal and results. Requiring a minimum amount of cognitive and conscious efforts in order to navigate the tasks is also a sign of simplicity. Limiting options on an interface to reduce the clutter is a step towards simplicity as well.

Consistency

Consistency is also very important for the goal of creating an invisible interface. If an interface constantly changes how it's functions are executed, interface become no longer consistent and evaluation becomes difficult to users. Consistency helps users to complete new tasks without having to learn a whole new toolset. It limits the number of ways actions and operations are represented. Users tend to apply their experiences with other interfaces with new interfaces they come across, this may confuse users if the consistency is not present. A user may feel like they need to gain new knowledge in order to achieve their goals and defeats the purpose of an invisible interface.

Dicoverability

Discoverability is very important in my opinion. If a user doesn't know about the commands to be called or buttons to be pressed, design fails the user by making him wonder about what actions to follow. Discoverability can be achieved in a number of ways. Size, colors, fonts, the shape of an element, an order which items on an interface are placed can be leveraged to make functions more discoverable. The interface can be made invisible if the user can quickly figure out what actions to perform by without spending too much time having to think.

"Tolerance" is a principle that understands the context which users exist outside their interaction with the interface. I use tools to manage NoSQL databases as a part of my job. If I drop a database or a collection(table) by accident, an option to recover them would make my job easier and less stressed. If the principle of tolerance is not present in the design of the tool, I will have the cognitive and conscious load of being extremely careful about some of the tasks.

"Ease and Comfort" principle can be used to understand the user outside of the interface they interact. If the users are not comfortable in their environment while interacting with an interface, like when they have to perform movements and gestures that make the task uneasy, the interface should be changed by following ease and comfort principle. Smaller buttons where the thumb is too big to be pressed, extremely smaller text which is harder for an average person to read, interfaces with too small or too big dials are a few examples where the principle of ease and comfort is ignored.

Question 2

Command line interface is one of the tools that I use daily to log in to remote cloud servers for various tasks. CLI is just a blank interface with minimal tolerance for errors. CLI will execute whatever the command that we enter regardless of the impact of the command. It doesn't warn the user before executing commands most of the time. I might want to execute a command which I found online to achieve some task, but it may corrupt some other settings. Since there are no warnings, it is difficult to decide whether to go forward with executing the command. Penalties for entering the wrong command could be severe. It may wipe out an entire directory or change the network settings and the user may not be aware of the changes until some other functionality is broken. Tracing the root cause for changes is extremely difficult in CLI. Committing errors in CLI is easy. All you have to do is typing commands without researching more about what they actually do. Reverting the changes can be daunting through CLI. There are no indications of what tasks to perform on CLI, previous knowledge is required in order to perform a task. Users could accidentally type some commands which alter the state of the operating system and left without options to undo their actions.

Constraints can be used to prevent users from executing commands by not allowing users to just type them and press “Enter”. The user should be alerted before executing a command about the impact of the command. Having a warning message followed by “Yes”, “No” options might prevent users from executing commands that could cause damage.

Improving mapping in CLI may require redesigning into an interface with options. Giving users an option to preview the results before executing commands would prevent them from committing errors. Heavily highlighting irreversible actions and mapping them with links to more information about alternative actions could also prevent users from executing error prone commands.

In order to improve affordance for CLI, it must change the appearance to GUI. Providing buttons with options for some tasks can drastically reduce user errors. Using colors to display the impact of the actions (red for higher risk, green for lower risk) indicates the affordance. Having options to undo actions, starting and stopping various processes, monitoring running processes in the form of buttons could also be considered as improved affordances.

Question 3

I remember playing Call of Duty PlayStation game with a friend of mine sometime back. I don’t play games often, because of that using the controller was challenging for me. I experienced slips when making characters stand and squat using the controller. To make the video game character stand, I had to pull the left stick backwards and forward to make the character squat. I keep getting them mixed up while playing the game even though I knew what I had to do. I was pulling the stick forward to stand up and backward to make the character bend over. Some people get used to the setting after playing for some time, but as a beginner, I would like to have the settings in a way that I am comfortable. There are options to change the controller settings, but its time consuming and confusing. Implementing a simpler interface to change the controller settings would be a solution. I had watch youtube videos of people explaining how to change the controller settings. This can be avoided with a better informative interface.

I made a few mistakes during the same time when playing Call of Duty. When changing the weapons, I was pressing buttons all over the place. I had a wrong mental model of how the buttons of the controller works. Time given is limited and the enemy gains the advantage if you spend more time doing some activities. Changing weapons and other controls such as jumping and running should be executed swiftly. I had to scroll through a bunch weapons which I wanted to use during the game and ended up using wrong weapons. Having to use controller buttons was a little frustrating. Players make these mistakes due to the time sensitivity of the situation. Late actions late may cost the game advancements. In this scenario, having voice commands would have been really helpful. If I could change weapons or ammo by giving voice commands, confusions can be avoided easily and actions can be executed quickly.

There are many changes that can be done to make a game challenging. Increasing the speed of the environment, requiring calculated moves and strategic planning are a few. Having required strategical planning would make the player think and learn to be creative in executing the tasks that the game required.

Question 4

An interface that I would see as a good representation of its underlying content is the lamp that I have on my desk (Shown in Figure 1).

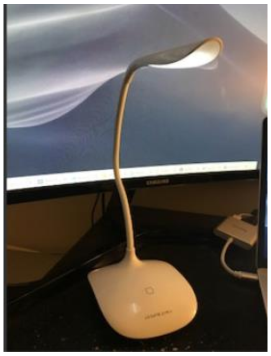
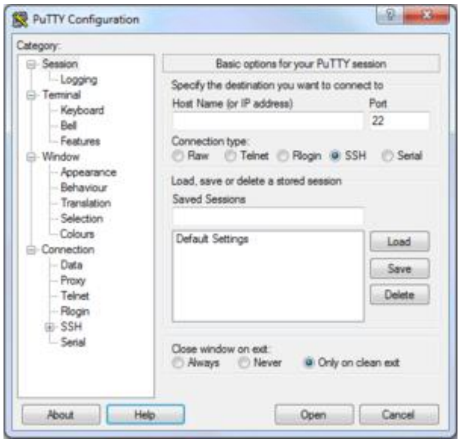


Figure 1. Desk Lamp

It is simple to use. The light comes on when I tap on the surface button (there is the sign of a square to imply where the button is). Tapping one more time on the same button will increase the brightness, it works up to three taps. One more tap on the button will turn the lamp off. It is a good representation because it doesn't give me extraneous details such as the how much power it uses or the wattage of the bulb is, and it fulfills the sole purpose of providing light without any complications.Having only one button to increase the brightness, turn on/off shows the simplicity of the design. I know that pressing the only button available would complete the three tasks mentioned earlier.

An interface that doesn't use a good representation would be "Putty" interface. Putty is used to connecting to remote Linux machines through any Windows computer. It requires an IP address and sometimes additional detail depending on the security settings of the server that user trying to access. It will open up a command line interface when you are connected to a remote server. It has a lot of details on the interface which I rarely use.



Host Name (or IP address)

Port 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 1. Putty Interface.

There is a mismatch on how most functionalities are represented on the interface. "Load", "Save", "Delete" buttons doesn't represent what they actually supposed to do unless you read about them or being told by another user and might develop a wrong mental model. We can save server details for later use on putty, making the life of the user easier by not having to type them every time, but the process of saving requires some learning curve. "Putty" is a useful tool overall. But it has represented most of the functionalities in a way that is difficult for users to figure out what inputs needed in order to accomplish their goals. The layout makes it difficult to figure out tasks such as where to upload a security key file. IP address of the server needs to be typed in the “Host Name box” and “open” button have to be pressed. Natural constraints are not exposed since the “Open” button is place among many other buttons and other information.

References

1. https://www.interaction-design.org/literature -
2. Udacity lectures.