Assignment P2
CS6750 Human Computer Interaction

Sunil Rao
srao374@gatech.edu

1 QUESTION#1

1.1 First Principle & Justification - Discoverability

Relevant functions must be made *discernable* by the user that let them identify actions required to perform to achieve their goal easily. Put in other sense, significant cognitive load could be taken off the user by making functions more discoverable thereby rendering the user to focus just on the task.

Gulf of Execution – Adequate needfinding methods can highlight what functions are most important & how to make them discoverable easily. This will help users identify actions to execute their intentions faster thereby bridging the gulf of execution.

Gulf of Evaluation – Users must also easily understand if their actions did change the state of the system. Functions actuated by the user must also indicate to the user what changed; for ex, me bullet pointing this paragraph on MS word by clicking on the bullet button would depress the button indicating I did click on it & place the paragraph as a bullet point instantly. I now know what I did & what happened as a result of that, thus bridging the Gulf of Evaluation.

1.2 Second Principle & Justification - Simplicity

Not throwing a pile of functions at the user even though it may be discoverable distracts or even burdens the user's cognitive capacities. The camera interface on iOS, makes a few key functions on its opening screen *prominent* like Capture button, Mode selection, Zoom, Camera Switch & *hides* second level of operational functions like timer, flash, contrast/exposure as options. This enables users to perform the most important aspect of their intentions viz. capturing moments as photo/video with ease.

Gulf of Execution – Minimalistic design allows users to access important functions easily & achieve their most salient goal instantaneously. All that the users would need do to capture an image is click on the camera app & simply hit the capture button – through simplicity of its functions made available by the interface. This bridges the gulf of execution significantly & renders the interface invisible.

Gulf of Evaluation – Not displaying a whole lot of information after user action, lets users sufficiently know that their goals have successfully been achieved. In this case, the image capture simply freezes briefly & disappears into the gallery thereby letting the user know that their operation was successful. No fancy system output or details thrown at users here.

1.3 Third Principle & Justification - Consistency

The principle establishes familiarity for users to enable easier transition between interfaces i.e., operating on new interface the same way they used to operate on other platforms. This offloads a huge chunk of their cognitive engagement as the need to learn how to use the new interface diminishes & makes the interface invisible in the prevailing context. For ex, images show the buttons on a physical camera & buttons on iOS camera. Notice how the buttons for image & video capture on iOS camera app are so similar to the physical buttons on the Sony camera. Users can quickly identify how to achieve their tasks of capturing images or videos on iOS, using their experience of using physical cameras, not having to learn the iOS interface, thereby bridging both the gulfs, making the interface invisible.

1.4 First Participant Principle & Justification - Flexibility

This principle allows for an interface to adapt to users' limitations; physical or environmental. iOS's accessibility setting options is a good ex of the usage of this principle. We will limit the screen size "*Reachability*" & "*Type with one-h&*" options for this discussion.

With smartphones' physical size increasing to provide larger screen real-estate to users that have been performing more transactions than ever on their phones, it becomes difficult for users with smaller hands to operate these devices with ease. iOS provides a "Reachability" option to lower the top half of the screen so it's within easy reach of thumb. Likewise, it provides an option for users to "Type with one-hand" to supplement to not just users' physical limitations but also to aide usage in contexts outside their interaction with the interface. I find these options very useful in both, the original context it was designed for & the surroundings impacting my usage context; 1) My hands are smaller & do sometimes use the “reachability” feature when I need to check on notifications. 2) I use the one-hand type option when I’m walking or entering addresses on maps when I’m in the car with the phone docked. The principle thus aides in reducing errors while using the interface bringing about better motor system engagement.

1.5 Second Participant Principle & Justification - Ease

This principle allows for a user to avoid fatigue or associated physical exhaustion in using an interface. We will talk about the *Speakerphone* option in smartphones to detail this principle.

With increasing phone sizes, it becomes cumbersome to hold a rectangular block up to our ears every time to talk. It adds fatigue to hands, neck & back. Granted that people are moving to Bluetooth headsets but it still has its own fatigue constraints to ears. The Speakerphone option allows for the user to talk hands-free thereby adding significant ease of interface usage to communicate in this context. This feature expands the horizons of the context it was designed for; users can use speakerphone while walking, cooking, exercising & adds to degrees of usability in multiple contexts. Usage of this principle in this context is to let users hold the device at their ease, knowing which button activates speakerphone & that they do not have to interact with the screen at all when in speaker mode.

2 QUESTION#2

2.1 **Interface:** *Traditional Gas/Diesel pumps at Gas Stations.*

2.2 Interface Problem Context

Almost all Gas stations in the US have multiple fuel options available – Diesel, Gasoline (87, 89,91 etc.). While there usually is some color coding to differentiate between gas options, they are **easy** to bypass users’ knowledge & attention to cause critical mix-ups like lower octane gasoline in higher octane vehicles, gasoline in Diesel vehicles, Diesel in gasoline vehicles. Reference image in Appendix.

2.3 Interface Penalties

Faulty fueling has severe repercussions like critical damage to engines that will need overhaul by mechanics like replacement of fuel pumps, cleaning of the tanks etc. This is way too expensive for a user mistake unforced by fuel pump interfaces they use almost every day.

2.4 Interface Constraints

Physical constraints like varying the very look & feel of nozzles can help users identify if something is amiss just by holding & trying to operate with it. For ex, Diesel nozzles are larger & cannot be inserted into Gasoline filler necks. Similar strategies could be designed to disable Gasoline nozzles from being inserted into Diesel tanks. RFID sensors on Gas dispensers that could detect fuel type from RFID transmitters on the vehicle & not dispense at all if there is mismatch, could significantly change the fueling system & completely avoid these costly user errors.

2.5 Interface Mappings

The color on the various nozzles could match color on the fuel tank openers. Diesel vehicles already employ this in some newer vehicles; their tank filler caps are green in color to match green diesel nozzles (usually). Likewise, all octane levels could have some color coding to identify compatibility. Also, there is very little information as to what these octane levels are in the first place. The interface could display on screen what each selection means upon respective button press to inform/alert/caution users.

2.6 Interface Affordances

Users can understand by merely looking at fuel selection buttons that they are to be pressed to make a fuel selection. But what if they want to cancel the fuel selection & change fuel type? This needs the user to start the transaction all over again as per the current design. Perhaps a pull capability to the press button could disengage the current fuel selection easily? Maybe yes, this could be a design option we could discuss. Also, signifiers could be used by playing an audio confirmation of the fuel selection. Since it is a button press anyway, which already perhaps beeps upon pressing, a computer-generated voice reading out the fuel type/octane rating could help user avert mix ups.

3 QUESTION#3

3.1 Game Selection: TempleRun on Smartphones/Tablets

The gist of this game is to control an explorer to help him escape from creatures that appear to chase him & keep him running as long as possible while collecting gold coins throughout. The controller must navigate the explorer Left or Right by swiping left or right on the screen & help him avoid obstacles by Jumping over or sliding under them, by swipe upwards & downwards respectively.

3.2 Game Slip

Slips occur in this game when users want the explorer to turn left but swipe right on the screen instead or when users want the explorer to jump over an obstacle but ends up swiping down leading the character to slide & derail from the system expectation.

3.3 Game Slip Rationale

As the explorer approaches a change in direction (like a T junction) or an obstacle like a tree branch or fire or moat, users must react quickly to actuate on screen as needed. Here users, being pressed against the apparent pace & time stress the interface induces, snap their h&-eye-brain co-ordination, & end up employing wrong swipe actions.

3.4 Game Slip Improvement

This game also uses device's gyros to gauge horizontal movements & embeds a function where users could tilt the device left or right to collect gold coins while not letting the explorer veer off course or change directions. One of the potential improvements could be to provide users the flexibility to customize controls to navigate the explorer. If users could rather tilt the device left/right to navigate left or right instead of swiping left or right, there could be decreased swipe activity & reduction in cognitive load associated with too much swiping. There would still be same swipe actions to collect coins but not collecting coins would not end the game whereas failing to change directions would.

3.5 Game Mistake

A mistake that occurs is the users' inability to judge the time & effort it takes between a swipe action & resulting activity the character in game makes. For ex, if the explorer approaches an obstacle, ambiguity around at what exact point must the user swipe upwards or how big of a swipe action he/she must perform remains. If the gap in a moat obstacle is larger, the user must incorporate extended swipe upwards action or if the gap is small & there is a fire obstacle right after the gap, a high swipe up action would l& the user in fire directly. So, this ambiguity often leads users to commit mistakes & lead them away from their goals.

3.6 Game Mistake Improvement

The game interface could perhaps offer a tutorial arcade that would also let users calibrate their swipe actions on screen to tie to appropriate activity in the game. This would allow users to subconsciously record the magnitude & precision of their finger swipe action & map that to relevant activity in the game.

3.7 Game Challenge

Slips discussed here are not necessarily due to a glitch in the interface but rather one of the very intents of the game itself. The game emulates high paced running action & requires good reaction time & motor system engagement from the user. The challenge posed by the game interface is to check users from running too long whereas the user tries to beat this interface's challenge by deeply engaging their motor system & reacting to direction changes & obstacle evasion.

4 QUESTION#4

Limiting the scope of this discussion to controlling a Chandelierlight fitting (with 5 bulbs) with Physical Electric Switch as the good interface & FEIT electric Smart-Bulbs controlled via FEIT app on iOS as the bad interface.

4.1 Representations of the Good interface: *Physical Light Switch*

4.2 Connections of the Good interface

A physical switch indicates 2 rigid possibilities – OFF or ON. If it is in OFF position, even if users are not familiar with what is OFF position, simply flipping it will indicate the change of status & vice versa. Even when multiple bulbs are connected to a light fitting that is operated by this Single Switch, users can understand or change their statuses by toggling just this one switch's positions. In case bulbs in a light fitting are not switching on despite toggle, users can arrive at the point of resolution rather quickly – either the bulb is worn out or there is some electrical issue.

4.3 First Criterion of the Good interface: *Making Relationship Explicit*

A clearly designated light switch maps easily to the device it controls. In our case, the switch to control Chandelierthat is clearly designated, using a label or general positioning convention under-st&ing, changes the status of the light fitting with a simple physical toggle. The relationship of the switch to the bulb is explicit – The switch controls the bulb(s). Even when there are multiple bulbs connected to operate via one switch, a simple physical toggle changes the status of all the bulbs & if not, the user can arrive at the point of resolution very quickly.

4.4 Second Criterion of the Good interface: *Exposing natural constraints*

Usually, switch flipped down indicates OFF position & flipped up indicates ON position. By the very affordance it imparts, a switch naturally restricts user operation to only 2 possibilities – ON or OFF. The only human error here is to not initially know what OFF or ON positions is, which gets registered over time. The design of this switch naturally restricts human errors by providing direct mapping of its functionality. Additionally, users can add signifiers to this affordance by labelling ON & OFF above & below the switch. This will help constrict user error even better,

4.5 Representations of the Bad interface: *Feit Electric's Smartbulb iOS App*

The app enables users to control their Smartbulbs via smartphones from anywhere providing remote on/off, dimming & color changing features.

4.6 Connections of the Bad interface

The interface does not capture SmartBulb feedback accurately. Here are some scenarios & their inferences that establish the notion of bad connections. To provide context into these scenarios, the app provides a way for users to logically group related Smartbulbs, for ex, I have 5 bulbs in my Chandelier& to control all at once, I created a logical Group device "Ch&elier" & added the 5 connected Smartbulbs under that.

Scenario#	**Physical Switch Position**	**Light Status**	**Individual Bulb status in App**	**Group Device status in App**
1	OFF	OFF	ON	ON

Scenario#	Physical Switch Position	Light Status	Individual Bulb status in App	Group Device status in App
2	ON	OFF	OFF	ON
3	ON	ON	OFF	OFF

Scenario#1: With the physical switch OFF, the app does not display real-time status of the smartbulbs & still shows they are ON in the app. This is a problem because if users have turned the physical switch OFF accidentally & are trying to control lighting remotely, there is no way for users to know if lights are indeed being switched on or off.
Scenario#2: This happens when one or more of the Smartbulbs in the Group (Chandelier in this case) have some issues unknown & do not switch on. The app indicates their individual bulb statuses as OFF but does not indicate why & the GROUP the bulbs are part of does not indicate any anomalies & still shows as ON even if all bulbs fail. This is incorrect representation.
Scenario#3: This happens when users MOSTLY operate Smartbulbs as regular bulbs when they are in proximity but use the app when they are away from the device/apparatus. Their statuses are not reported accurately real-time on the app if app is not used to control them.

4.7 First Criterion of the Bad interface: *NOT Making Relationship Explicit*

It is not as straightforward as in regular physical switches to determine the state of a bulb explicitly. In other words, it is hard to immediately tell why lights are not on – is the physical switch on or off? If it is on, is the app showing the bulb as off or on – the varying parameters to determine the status almost multiply. This happens more gravely so because the app does not apply real-time 2-way feedback with its connected smartbulbs as indicated in the section above.

4.8 Second Criterion of the Bad interface: *NOT Exposing natural constraints*

In trying to emulate physical switches, the app does not really provide similar affordances in its representation of smartbulbs as buttons to start with. There is no definitive way to tell if the light is really ON or OFF even if the buttons signify ON or OFF. This potentially urges human error in the way of their understanding of whether the light is actually on or not.

5 APPENDICES

5.1 Reference Image 1

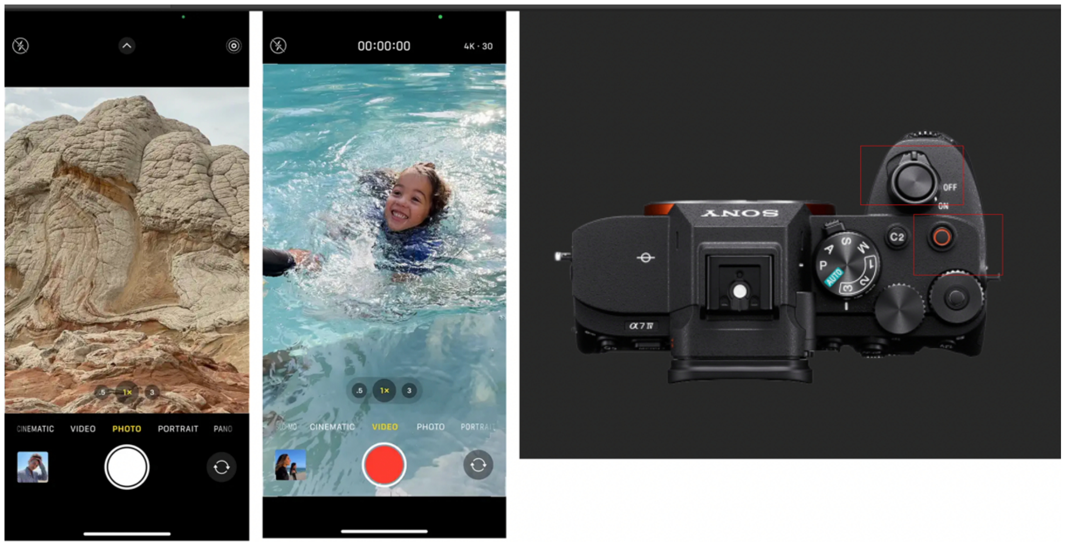

Figure 1 — (L-R) iOS Image capture button, video capture button, Image/Video Capture buttons on physical Sony Camera

5.2 Reference Image 2

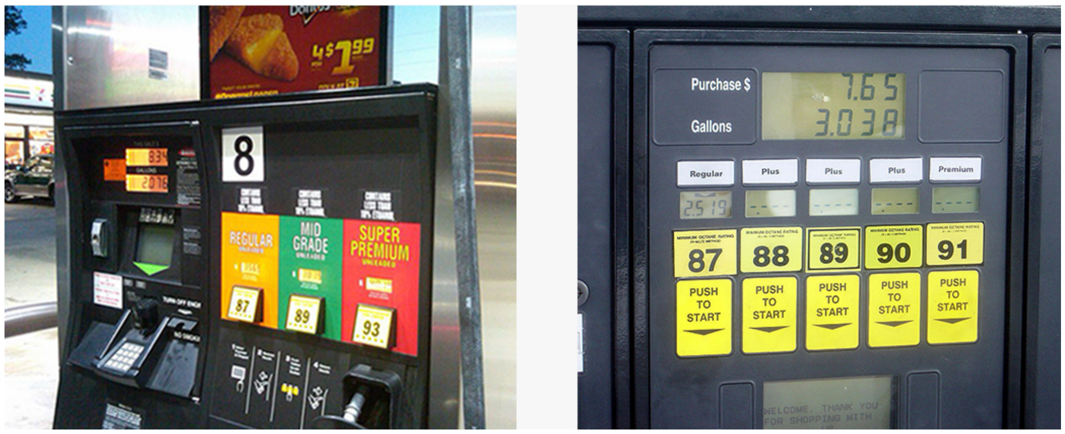

Figure 2 — Gas pump at Gas Stations in US

5.3 Reference Image 3

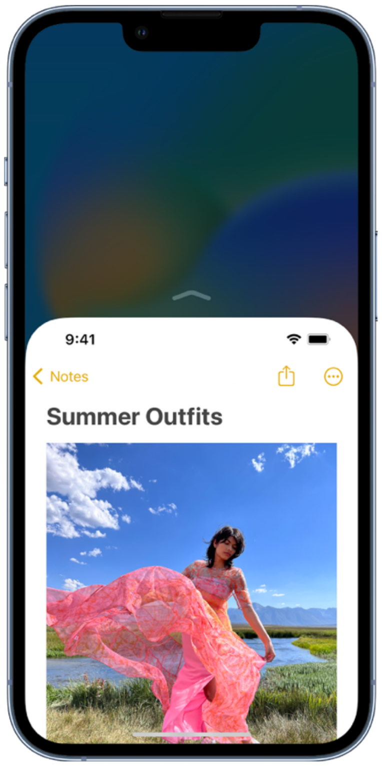

Figure 3 — iOS reachability

5.4 Reference Image 4

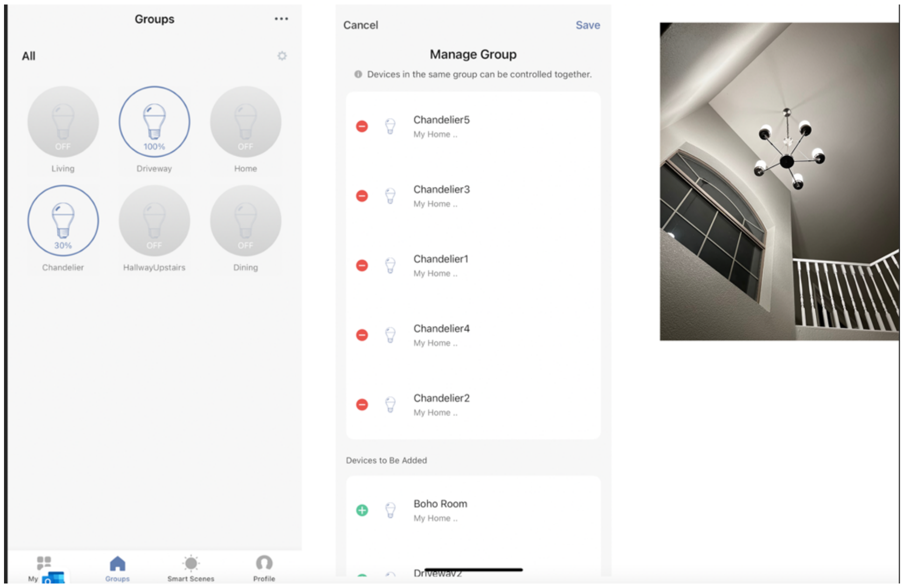

Figure 4 — FEIT app showing the Group Devices, Individual bulbs and the actual light fixture.