Assignment-P3

Arti Chauhan
achauhan39@gatech.edu

1 QUESTION-1

1.1 Principles – Invisible interface

- **Consistency** : Consistency principle promotes reuse of user's existing knowledge. In other words, Consistency allows user to readily transfer expertise with one system to other, with little to no effort. This enables user to quickly map his goal into actions and execute it, there by shrinking gulf of execution. Eg : Many text-editors use Ctrl-c to copy text. This enables user to perform task of copying, without spending much time on figuring out actions required to complete the task. By being consistent in context and meaning within and across interfaces, user's learning-time is reduced significantly, and users arrive to the point of Invisibility much quickly.

- **Feedback** : Principle of Feedback advocates providing *immediate* and *relevant* feedback as user executes an action. Clear and concise feedback enables user to *interpret* and *evaluate* impact of his actions and current state of the system. It allows him to adjust his course quickly and move towards his goal , making interface invisible. Thus feedback, when implemented correctly, greatly shrinks gulf of evaluation. Eg: When trying to set desired temperature, Nest immediately gives user feedback on what the current temperature of room is and how much time it will take to reach desired temperature. This helps user evaluate if his task was accomplished or not.

- **Discoverability** : Principle of Discoverability emphasizes on making objects, actions and options visible in the interface and removing extraneous information. Discoverability enables user to identify what actions are possible and how to perform them, helping him cross gulf of execution swiftly. By removing extraneous information from the interface, we minimize clutter and steer user's focus on actions/buttons that are key to accomplishing his task. Eg : a) The very shape of Nest (as a dial) tells user

that it is meant to rotate. b)Presence of a looking-glass icon on an interface tells user that he can perform a search. c) Many applications organize related functions under one menu-item for easy discoverability.

1.2 **Principles - Participant view**

- **Tolerance** : Tolerant interfaces allow user to recover from their mistakes without penalizing them. This principle takes into account user's context and expects that there will be scenarios where user will make mistake (maybe because he doesn't know the system well enough yet or due to slips) and provides a clear way to recover from it. A well-designed interface not just displays an error message when a mistake is made, it predicts where there are higher chances of user making a mistake and eliminates those conditions in first place. Eg : Auto save feature in MS-word predicts that there might be scenarios where user accidentally closes the document with out saving or application crashes. Auto-save helps user recover his work.

- **Equity :** Principle of Equity allows interface to take participant view by accommodating users of diverse abilities, preferences and cultural background. Interfaces that leverage manipulations ingrained in basic human behavior build Equity amongst users, irrespective of their technical-skill, culture or language. Eg : *Scissors* icon in a text editor readily conveys to user (of any language) that it is meant to cut a text. Pulling down a menu or flipping a page in e-book is intuitive to both novice and expert user and hence builds same experience (ease of use) for users with diverse technical expertise.

2 QUESTION-2

2.1 Intolerant Interface

The days when I telecommute, I remote access my Company's intranet via a VPN UI. To access the intranet, user must provide a soft-token#, which is generated using user's password and is valid only for a short period of time (1 minute). This VPN software is very intolerant to user errors.

If WiFi happens to lag a bit , token generated by Token-Generator doesn't get sync'ed quickly enough and VPN UI rejects user's passcode. Another case of rejection is when user enters the passcode close to 1-minute passcode-expiration boundary. It is very easy to commit these errors, as in both cases, user enters the right passcode, but system perceive it invalid due to time-lag, which user has not clear way to detect.

Three wrong retries, and it freezes all of user's accounts in Company's network , including access to different DBs, portals and tools and an email is sent up your chain of command. To unlock, user has to file request for each account separately. In other words, recovery from this error takes 3-4 days.

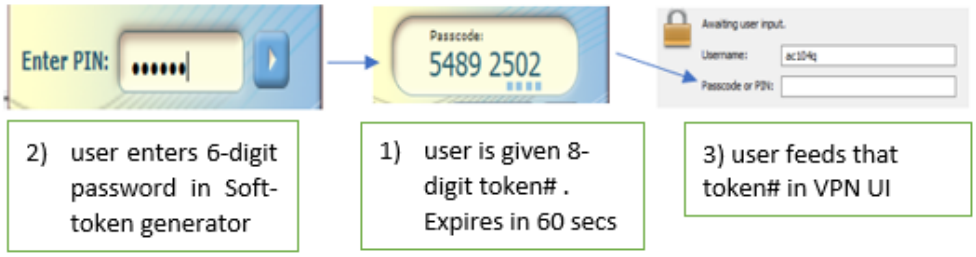


Fig 2a : process of accessing Company's VPN

2.2 Redesign

- **Leverage Constraints** : When user enters wrong passcode, an error message is shown that passcode was invalid, and user should try in 60 seconds. However, UI allows user to enter passcode before 60 second window expires. Three wrong retries, and hell breaks loose. For an error with such severe penalties, interface should eliminate error-prone conditions in first place. One way this can be achieved is by providing a constraint in the interface where user can't enter passcode in VPN UI until 1-minute window has expired. This also offloads the task of keeping track of expiration-window from user to interface.

- **Improved Mapping** : Improved Mapping allows creating interface that makes it clear to user what the effect of their actions are. Though VPN interface throws an error at when passcode is rejected, but I would argue this not enough for user to determine all consequences of his action.

Interface doesn't communicate to user that after 3 wrong attempts all his accounts will be frozen. This can be overcome by displaying on the window number of wrong attempts user has made so far or conveying this information graphically with a color bar with 3 slots (each get red color if password was not accepted). That way user slows down as he is about to reach the max limit of 3.

- **Improved Affordance** : Affordances, by their very design, tell the user what to do. Interface should gray out passcode input field when user's passcode is deemed invalid by the system and display how much time is remaining in 1-minute expiration window. Grayed out input field tells user that he needs to wait and *Time-Remaining* tells for long to wait before he can re-attempt.

3 QUESTION-3

3.1 Game description :

Temple Run is a 3D endless running video game (by Imangi Studios). The game controls an *Explorer* who obtained an ancient relic and is running from evil demon monkeys who are chasing him. As the game is an endless running game, there is no end to the temple; the player plays until the character collides into a large obstacle, falls into the water or is overtaken by the demon monkeys. When player needs to turn left or right, the touchscreen can be swiped in the corresponding direction. If the player wishes to jump over an object, the screen can be swiped upwards. If the player wishes to slide under an object, the screen can be swiped downwards.

3.2 Slip :

When I started playing this game, I sometimes made errors in choosing the right direction and/or action - I knew I am supposed to *go right* but I *go left*. I knew I am supposed to *jump* but I accidently *slide*. These errors were sometimes result of rapidly changing course but most often , me not paying attention.

Redesign:

- As course takes twist and turns, interface may throw a cue to what action to take, which will constraint users, especially beginners from making

slips. One way to quickly recover from such errors is to introduce *check-points* from where user can restart the game if he fails an obstacle , rather than starting the run from very beginning.

3.3 Mistake

I can see that I am reaching an obstacle that will require me to jump or slide. I attempt to jump or slide depending on obstacle type but fail. why ? because game expects you to make the jump or slide at a specific distance from the obstacle. Too early or too late, you end up in ditch. For me as a novice user, it was hard to quickly determine *window of perfect jump/slide*. To make matter complex, this *window* is different for different types of obstacles.

Redesign:

- Above mistakes happen as I don't have a clear mental model of optimal distance from where jump/slide should be made when an obstacle is encountered. One enhancement that can be made in the interface is to light the path with a different color when Explorer is about to run into an obstacle. Eg : color the portion of path green which represents time-window of perfect jump/slide. That way user has a better representation of when to perform an action.

3.4 What Makes Game Challenging

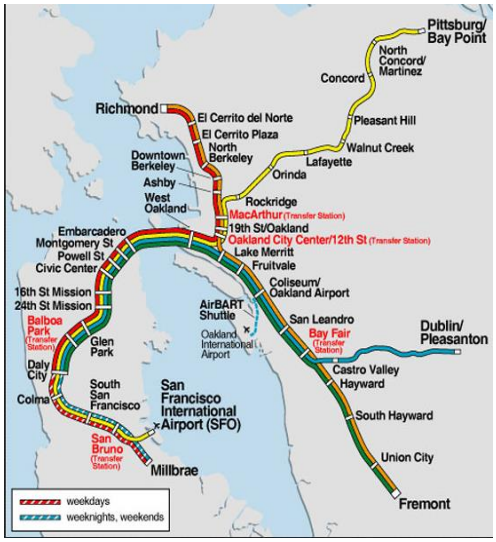
As the game progresses, course becomes more and more difficult. At that point , *Explorer's* survival rests on player's reflexes and instincts. As course takes more abrupt turns, players has to decide in split-seconds whether to go left or right . Obstacles would turn up in most unexpected places, so player has to quickly decide whether to jump or slide or deflect. Errors made here are not slips or mistakes but due to his inability of being extremely quick and responsive. This makes game fun and challenging.

4 QUESTION-4

4.1 Good Representation

Figure on below shows a snapshot of metro rail transit app, which I used quite often. This uses a good representation of underlying data.

1. **Excludes extraneous details** – this representation removes irrelevant information such as topology of underlying areas and distance between cities, as it is not required for me to accomplish my goal of going from city A to city B. All I need is different options (rail lines) available to me between two destinations.
   By not displaying distance and topology, interface is more clear and clutter free and helps user focus on important task of choosing the correct train.



2. **Makes relationship explicit** : This representation uses the color which matches the name of the train-line Eg: route of Blue-Line is represented by blue color , Red-line by red color and so on. Laying things out like this makes it easy to tell for which route I have multiple options vs route with single option. By using simple color-coding (black vs red) for text, it makes the relationship between stations explicit - cities are marked with red-font if it's a transit station, else it is marked with black font. It also brings out relationship between different train options in temporal domain – it conveys which trains are available on weeknights/weekends vs weekdays (Eg: between Daly City & Millbrae) by using dashed color instead of solid fill.
   I would argue that layout of white lines (used to mark a city) in this representation brings out relationship in spatial domain as well – this layout gives user an idea about relative distance between any two given destinations.

4.2 **Not so good representation**

At work I deal with huge amount of data that captures user experience KPIs (key performance indicators) across US on a mobile network. For this

purpose, my organization heavily uses Excel, which works great to find any anomalies or trending but poses a significant obstacle in performing meaningful analysis when this data has positional information (zip code, city or latitude/longitude). A scatter-plot or bar chart options presented by Excel is not the effective representation of underlying data.

Violation of good representation criteria

1. **Doesn't make relationship explicit** : With Excel's pivot tables, I can't determine the relationship between performance of KPIs in one area vs another. In contrast, we if view same info in Power-BI, it understands latitude/longitude or zip code/state/Country and automatically projects relevant KPIs on a map. This makes the relationship between KPI value and underlying morphology of an area. Eg : Dense urban area have higher call-drop rate than sub-urban areas.

2. **Doesn't expose natural constraint** - In pivot table, Excel interface allows user to put in latitude/longitude on x or y axis, which by default show average of latitude/longitude. It fails to convey the constraint of underlying positional data that mean/median operations on it are meaningless. In contrast, Power-BI is good at understanding this constraint and prompts user to select a map visualization and grays out line/scatter/bar charts.

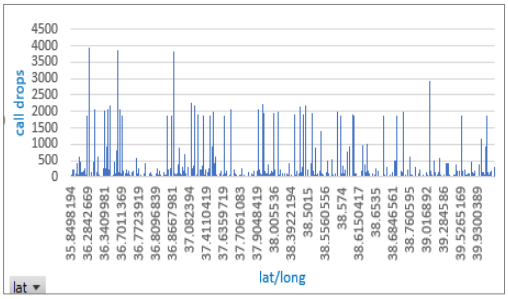


Fig 4.2a : positional data in Excel

Fig 4.2b : same data in PowerBI

Effective visualization of data is a key tenet of good representation, which Excel fails to handle for positional data.