Assignment P3: CS6750 Spring 2019

Dan Higgins
Daniel.higgins@gatech.edu

Design Principles and Heuristics (from Lesson 2.5)

Question 1

Describe how Discoverability, Simplicity, and Mapping design principles might be used to support the creation of an invisible interface?

An invisible interface is an interface that the user does not have to spend any time thinking about instead all their cognitive resources are allocated to performing the task at hand. **Discoverability** at first may seem counter to this point of an invisible interface as Discoverability "is making functions visible to the user so they can discover them rather than learning them elsewhere."(Joyner, 2017, Lesson 2.5) However, what invisibility means in this context is that the interface is so apparent that it can be ignored. And discoverability serves this purpose well by making tasks visible without distracting the user from that task at hand. Additionally, Discoverability supports novice and expert users by providing visible hints to more complex but potentially more efficient uses of the interface. (e.g., Visually showing keyboard shortcuts alongside a button for the action.)

Simplicity on the other hand often has a tension with Discoverability as Simplicity is about removing from view rarely used features. Simplicity, however, aligns well with the notions of invisible interface design as it advocates the use of a design that is easy to understand for users of all experience levels. Simplicity could be used to support the creation of an invisible interface by putting front and center only the interface items that are needed and only when they are required.

Mapping is the process of aligning activities in the interface with their effects in the world. Mapping directly aligns with our notion of the invisible interface because with a good mapping we can essentially ignore the interface and instead focus on the task we are trying to achieve. Nielson says mapping should follow real-world conventions and use words and phrases that are familiar to the user. Mapping is used to support the creation of an invisible interface in many web browsers by using terms like 'bookmarks' and 'history' they could have used more technical terms but these terms map well to their intended use.

Describe how Flexibility and Feedback design principles could be used to create interfaces that emphasize the participant view of the user?

"While the processor and predictor views emphasize only the interaction of some user and some interface, the participant view also looks at the interaction of both within the context of a larger system… so anytime you're looking beyond just the user and the interface, you're likely employing the participant view in some way." (Joyner, 2017, Lesson 2.1)
Flexibility advocates the user's ability to customize the interface in multiple ways to support executing the same task. The flexibility design principle could be used to support the user in different contexts, for example, customizing an audiobook application to behave differently when exercising as opposed to when simply sitting in a chair. Applying Flexibility in this way emphasizes the participant view of the user.
The **Feedback** design principle advocates keeping the user informed with immediate, unambiguous information on how the users' actions are interpreted and what changes they may have caused in the system. Thus when feedback is adjusted to better meet the needs of the user in different contexts this design is emphasizing the participant point of view. For example, providing haptic feedback when a cellphone is not on the person's body may not be as appropriate as providing audible feedback.

Question 2

Select an interface that is intolerant of errors the user commits. Describe the interface, and describe how it responds to user errors, highlighting how easy the error is to commit and the associated penalty.

The news aggregator App that I use on my Android phone has some excellent qualities. For example, I like the news sources that it applies to aggregate the new: NY Times, WSJ, Washington Post, My local newspaper, etc. However, the interface is often intolerant of errors. One error that I am prone to commit is pushing the back button to many times. Since the application aggregates from multiple news sources if you select an article to read you are taken to a screen that wraps the original source provider. From this screen, you may read the chosen item and additional articles from that provider. To exit you hit the back button to pop the articles, you have read until you get back to the main screen. However, if you hit the back button one too many times (one past the main screen), the application quits without warning. After reentering the App, you are taken to a new (refreshed news list) this list may or may not have articles that were on the list prior to the App exiting. It is also important to note that when you select a news article from the main window, it is obvious who the news provider is, as this data is shown as a trade make icon just above the article. However, once you click into that article, it is no longer obvious as you now must scroll to the top of the screen to see the provider info but by doing so you will lose your place in the article were reading.

I believe the application could be improved in the following ways. First by applying a **constraint** on the back button. Constraints as a design principle are limitations placed within an interface preventing a user from making an erroneous action. The constraint I would apply is simply popping a warning message if you hit the back button on the main screen. This would prevent the error of exiting the application by accident.

Next, I would add a visible counter or "**mapping**" showing the news source and a count of how many articles deep you are in a newspaper. Mappings as a design principle use the idea of following real-world conventions with semantics that are familiar to the user. This allows the user to easily "map" what is happening in the interface to what is happing in the real world. The visual mapping, for example, would convey the message I am reading WSJ, and this is the 3rd article. Because this mapping is natural and logical to understand the user is less likely to make an error.

Finally, I would add an **affordance**. An affordance as a design principle "affords" or suggests how an interface is supposed to be used. "In other words, an object with an affordance, basically tells the user by its very design, how it's meant to be used."(Norman, 1990) I would change the back button and replace it with a swipe widget which would include the above mapping and additionally the widget would show two corners of a newspaper and animate pages turning forward or backward until you return to the main screen.

Mental Models and Representations (from Lesson 2.6)

Question 3

The game I have chosen is "Pac-Man" in this classic game you are given three lives with the objective of navigating your Pac-Man through a series of mazes eating strips of dots and fruit that are placed in the maze for points. You must navigate the maze without having your "Pac-Man" trapped by one of the four ghosts in the game whose objective is to eat your "Pac-Man" (Being eaten takes a life). Each time you complete a maze by eating all the dots on that maze you are presented with a new maze, in successive mazes the ghosts get smarter, faster and more aggressive. You continue to play until all your lives are gone, the player with the most points (highest score) wins.

A **slip** in the game can easily occur when you are navigating the maze and want to move in a downward or upward direction but your "Pac-Man" continues to move straight instead. **Why** a player makes this *action-based* slip is often a problem with eye-hand coordination. The slip occurs because the player must hold a controller in the forward motion direction to keep the Pac-Man moving but must stop the forward movement to change directions in the maze. In this slip, your mind knows what you want to do, but your hand fails to do it correctly. **To prevent** this slip, a modification could be made to the game such that if the player in advance of the direction change simply indicted a downward or upward movement with his controller (i.e., an intent to go up or down) anywhere in the forward progress on the maze before the turn. The game would introduce a “wall” in the maze (*constraint*) to disallow forward progress of the “Pac-Man” past the turning point and thus force the turn without error.

A common **mistake** players make is to eat a low-value fruit when a high-value fruit is equally available thus giving the player lower overall points for the action taken. The reason **why** this mistake is made is that the points for eating the fruit (which are provided in the instructions) are not shown to the player until after they eat the fruit; this is a *knowledge-based* mistake. **To prevent** this mistake the value (*representation*) of the fruit could be revealed immediately when the fruit is on the screen showing the value for both fruits the player will correctly take the high-value fruit. In addition to showing the value just above the fruit, a list of all the fruits and their values could be displayed on the bottom of the screen as the game is being played making the points for eating a fruit easily *discoverable.*

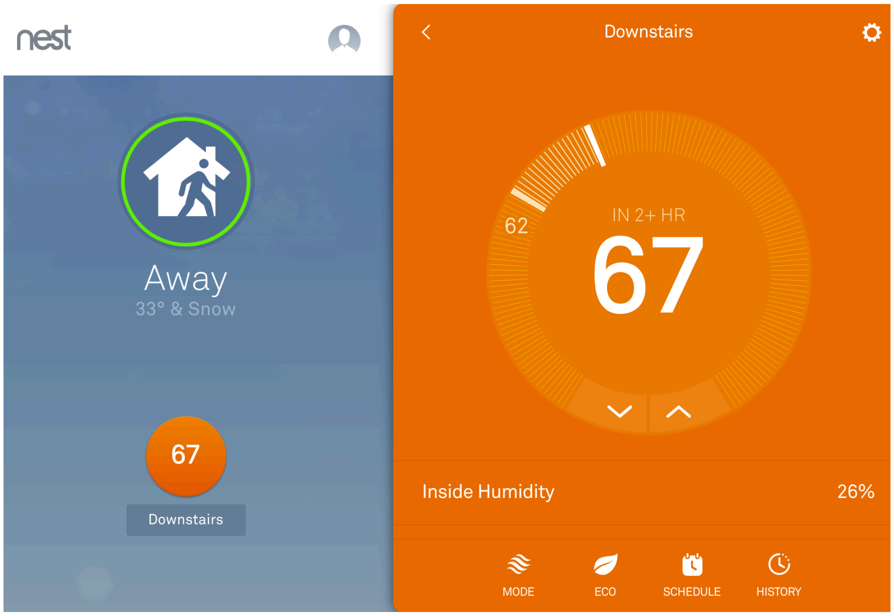
Question 4

An interface that I would argue uses a good representation is the web interface for the Nest thermostat. This interface provides numerous strong characteristics of a good interface representation. We look at two more closely here:

1. **It makes relationships explicit**. This is achieved by having a layout similar to what you see on the physical thermostat that is in the home. This web interface provides a display with the image of a thermostat that looks precisely like the thermostat in the house including all the current values that would be on the thermostat in the home if you were looking at it instead of the web site. (Current temp, desired temp, outside temp, etc.) Additionally, the interface behaves similarly to the device on the wall in the living room if you turn the gauge up the desired temp goes up.

2. **It brings Objects together with the representations between them.** The use of the up and down arrows on the thermostat dial make clear the relationship between these objects. One brings the desired temperature up and the other brings the desired

temperature down. The background color of the screen also provides information about the objects relationship with the overall heating system, moving the arrow up increases the desired temperature along with this action is a screen color change to orange. This color indicates a sense of warmth or warming while turning the temps down would produce a blue background indicates cool or cooling.



An interface that I would argue uses a poor representation is the electronic parking brake in my car.

1. **It fails to make relationships explicit**. The poor representation here is that the parking brake is a toggle button that you must either push or pull to engage or disengage the brake. The relationships between the brake and the toggle is not at all explicit. I have had cars with the parking brake on the floor to left of other operating pedals, in these cars it is obvious that you push the pedal to engage the brake. I have also had cars with a lever handle in between the driver and passenger seats with this lever it is clear that pulling will engage the brake. With the toggle button however it is unclear whether I should be pushing to engage or pulling to engage and even now after owning the car for a year I still often

toggle in the wrong direction.

2. **It fails to bring Objects together with the representations between them.** The poor object representations of the parking brake in the car is that even though there is an icon on the dash indicating whether the brake is engaged or not this indicator is not placed in any direct relation to the brake toggle. (You must visual search among numerous other indicator lights on the dash to find it) The brake indicator light is also not placed in any direct relation to the drive indicator lights (Park, Reverse, Drive, Neutral) Having a these indicator light objects in a good relation would be (in my opinion) the next most important relational representation as the car must not have the brake on if you are moving or attempting to move. Our solution to these poor representation problems is that we do not use the parking brake in this car unless we are parking on a hill.

References

Joyner, D. (2017). Human-Computer Interaction. Retrieved from https://www.udacity.com/course/human-computer-interaction--ud400

Norman, D. A. (1990). *The design of everyday things* (1st Doubleday/Currency ed.). New York: Doubleday.