Assignment P3:

Yasaman Vaghei
yvaghei3@gatech.edu

1 QUESTION 1

1.1 Supporting the creation of an invisible interface

To answer the first part of this question, I chose to discuss about affordances, mapping, and consistency.

1.1.1 *Affordance principal*

Affordance principal specifies that the design is required to hint at the way that it should be used. This is in line with the idea behind creating an invisible interface. Such interface allows the user to focus on the task rather than spending a great amount of cognitive load to figure out the way it needs to be used. For example, in an interface, the user automatically knows buttons are embedded to be pressed or knobs need to be rotated.

There are three stages in the gulf of execution: identify intentions, identify actions, and execute in interface. Following the affordance principal in the design helps with identifying actions in the interface. After identifying the intention, user tries to identify the actions required. At this stage, if the interface itself suggests the way to operate it, the user can easily move forward to the execution phase. For instance, visualizing the recycle bin's folder shape as a trash bin suggests that this is where the user can dump the files that they no longer need. Therefore, the user easily identifies that they can drag and drop the un-needed files to this folder.

Similar to the gulf of execution, gulf of evaluation also has three stages: interface output, interpretation, and evaluation. Following the affordance principal in the design helps with all three stages. For example, assuming the buttons on the mechanical keyboard, the user knows that they need to press those buttons to type specific letters. After executing the action in the execution phase, the user immediately knows the button is pressed since it moves down to the keyboard's frame. This is what the user expects immediately after pressing the key button since the design suggests the way the buttons on the keyboard are required to be used.

1.1.2 *Mapping*

Mapping principal forms a relationship between the vocabulary or signs that we use on a daily basis and what happens in the system. Following this principal helps making the interface invisible. The vocabulary or signs used to hint to how the system operates, or the consequences of the actions performed in the interface. This reduces the cognitive load on the user and helps focusing more on the task rather than identifying the consequences of the actions.

Following the mapping principal in the design helps with identifying actions in the gulf of execution. After the user identified their intention, they start thinking about what actions they are required to take using the interface. If the interface shows or hints to the consequence of the action, it becomes easier for the user to identify different actions and choose their desired ones. For example, take the hot/cold indicator on the faucet into account ("Figure 1"). On the knob, two colors (i.e., red and blue) are indicated on left and right, respectively. This is a simple mapping being used, meaning that turning the knob to the left shows higher temperatures while turning the knob to the right indicates lower water temperatures. This reduces the cognitive load on the user to identify the desired action(s).

Figure 1— Faucet temperature indicator [1]

Following the mapping principal in the design also assists with interface output visualization and interpretation in the gulf of evaluation. If the elements used in an interface are designed to be mapped to the consequence of the actions, executing the action, user can immediately visualize the output. Taking the same example of the faucet into account, after turning the indicator to the right (as can be seen in "Figure 1"), user can visualize the output. This means that the indicator is placed on the cold temperature as desired. Also, the color mapping helps the user to interpret the output.

1.1.3 *Consistency*

Consistency refers to have the interface design consistent with what was designed more broadly as a community. This can be associated with invisibility by learning. For example, the shape of the Windows start key has been consistent in different interfaces such as keyboards ("Figure 2") and Windows 11 screen ("Figure 3"). The users who are experienced with Windows and have used the Windows start key before (i.e., previously learned about the use of this key) can take advantage of the invisibility by learning.

Figure 2— Windows start key on the keyboard [2]

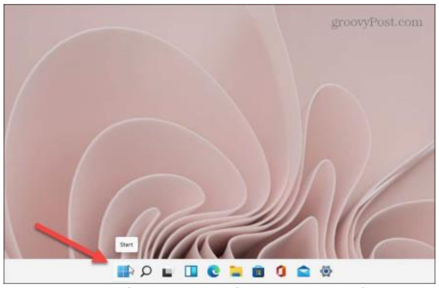

Figure 3— Windows start key on Windows 11 screen [3]

Following the consistency principal in the design helps with identifying actions in the gulf of execution. Once the user is aware of the intended action, it is easy to find the action required. This also reduces the time and effort the user spends in the gulf of execution.

1.2 Emphasizing the participant view of the user

To answer the second part of this question, I chose to discuss about flexibility and equity. A flexible design considers users with different levels of expertise and preferences. Flexibility of the design is helpful in different contexts. For instance, many software development kits (SDK), which are complicated pieces of codes, come with a graphical user interface (GUI). While novice users employ the GUI, experts prefer to use the SDK. However, in the context of a quick test or being in a busy environment that increases the risk for error, expert users also use the GUI. Also, the equity principal enables users with different abilities to use the interface in different contexts. For example, people with mobility disabilities might use voice to operate their phone instead of pressing buttons like healthy individuals. However, depending on the context, healthy individuals might use the voice commands as well (e.g., the phone is in the user's environment but not easily reachable).

2 QUESTION 2

The interface that I would like to discuss is a standard ATM machine. This machine allows the account holder to manage the funds in their account. ATMs generally makes it possible to check the balance of the account(s), deposit and withdraw money, and transfer money to other accounts. For this purpose, users require to have their debit card inserted into the ATM machine. After that, it is required to insert the pin using the provided pin pad ("Figure 4A").

The pin pad is not shielded in many of the ATM machines across the United States and Canada. Most of the users forget to cover the pin pad when pressing down the numbers of their pin. This exposes the user's security to a great extent. For instance, skimmers may not only be in the right spot to see the pin being typed on the pin pad, but also set up cameras to record the pin as users enter it.

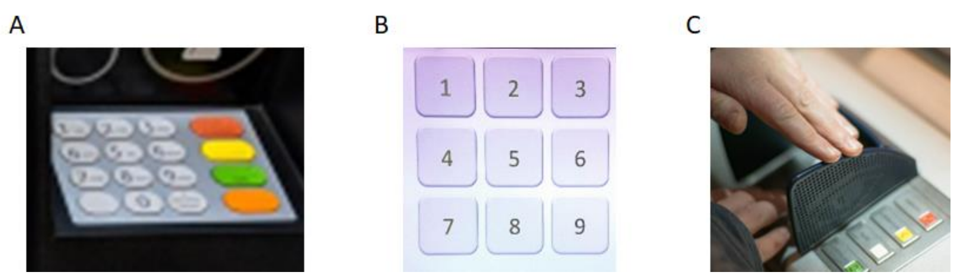

Figure 4— Pin pad designs for ATM machines

Constraints can be added to the interface for the purpose of avoiding this issue. For instance, the design of the mechanical pin pad can be changed to on-screen ("Figure4B"). The on-screen pin pad can get activated (i.e., light up to visualize the numbers) only when it is covered ("Figure 4C") using light sensors. After entering the correct pin, the on-screen pin pad stays activated without covering to complete the rest of the transaction requirements.

The use of affordance principal can assist guiding users to perform the desired action of covering the pin pad when entering their pin. Several designs were proposed previously to cover the pin pad (e.g., "Figure4C"); however, they only cover the sides of the pin pad. It would be beneficial to add a transparent shield within an optimal distance from the pin pad where the user can place their free hand to cover the upper part. On the transparent shield, a shape of a hand can be drawn ("Figure5"). This immediately hints the user to place their hand in the right place.

Improved mappings can be used to better re-design the process. For instance, blinking bar lights can assist the user have a better experience. First, the user is required to insert their debit card into the ATM machine. Therefore, a blinking light surrounding the place where the card is inserted starts working until the user inserts the card. Then, the user needs to cover the pin pad for the on-screen pin pad to get activated. The user first needs to place their free hand on the transparent shield. A blinking light with a distinct color (e.g., "Figure5") surrounding the place where the user requires to place their hand starts blinking at this time. At the same time, a message indicating the need to place the hand on the blinking area to activate the pin pad can be shown on the ATM machine's screen. After placing the hand in the correct position, the on-screen pin pad gets activated, allowing the user to enter the pin safely.

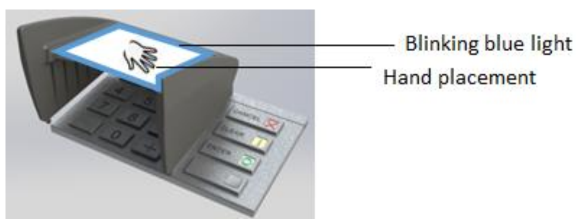

Figure 5— Pin pad re-design idea

3 QUESTION 3

Angry Birds is a video game in which the goal is to destroy the green-colored pigs with eponymous flock of angry birds ("Figure 6").

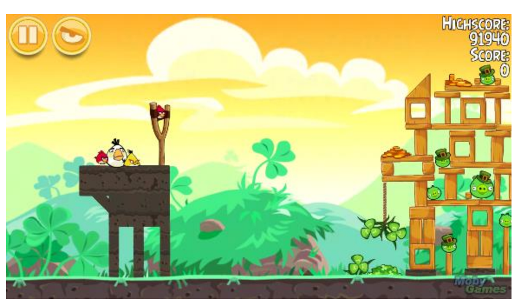

Figure 6— Angry Birds sample design

A slip that a player of the game might make is accidentally releasing the slingshot. This results in flying the bird on the slingshot at a speed and angle that is not desired. Therefore, the bird will not destroy the desired green pig(s).

There is one main reason as to why the player might make that slip and it is the fatigue in the index finger caused by holding it on the key pressed and releasing for a long time during the time of the play. This happens because for the slingshot to get activated and allow the user to change the desired tension in the string and change the target angle, one must hold their index finger pressing on the click button of the mouse for several seconds and then release the press to fly the bird. Repeating this procedure several times causes fatigue in the fingers and hand. Users may accidentally release pressing the index finger on the mouse to rest the finger. However, this causes a slip to happen.

In order to improve the design to prevent the slip in future, two different buttons can be designed: one for activating the slingshot, and the other to release the slingshot and flying the bird. This way, for instance, the user right clicks once to activate the slingshot. They then set the desired angle and tension in the string. They then left click to release the bird. This not only prevents fatigue in one finger, but also distributes the clicks between two fingers. Therefore, the risk of making that slip which is the result of fatigue in the index finger is reduced.

A mistake that a player might make can be flying a specific type of bird to a wrong target. As can be seen in "Figure 6", the types of the birds are different and each of them have a specific functioning when it comes to destroying the structure and the green pigs. Larger birds are good for short-distance targets and can destroy more elements, while smaller birds are great for destroying a target with a smaller diameter but further in distance. The player may mistakenly use a larger bird and try to hit a target at a further distance and vice versa.

The reason behind this mistake is the lack of information presented to the user. When a specific bird is up on the slingshot, the user does not immediately know what the function of it is. Therefore, the user may aim it at the wrong target.

In order to avoid this mistake, it is beneficial to visualize small pieces of information relevant to the functioning of the type of the bird on the slingshot on the screen at each stage. This way, the user would know aiming a larger bird to a further distance is a mistake and would try to come up with the right target for that specific type of bird.

There is an interesting aspect of the design of Angry Birds that makes it challenging even for the expert players. The challenge is that different structures are presented at each level with different types of birds being used to destroy the structure and the green pigs. There can be several ways to achieve the goal in this game for each of the structures at different levels. There is no obvious target for the specific types of birds to hit. One can pass a certain level using different strategies.

4 QUESTION 4

An example of a good interface design is the Cineplex seat selection map. This map is embedded into the Cineplex website and allows the users to select their desired seats in an auditorium to watch a movie. As can be seen in "Figure 7A", the screen is being shown in the front. The seats are presented exactly the same as the layout they have in the auditorium. Row numbers are shown on both sides. Different seat colors are shown to assist with choosing the desired seats. Dark blue colors show the available seating options. Seats shown with light blue color and disability sign are considered for disabled people. The seats taken previously are greyed out and are not selectable. Once the user selects their desired seat(s), the seat color changes to orange.

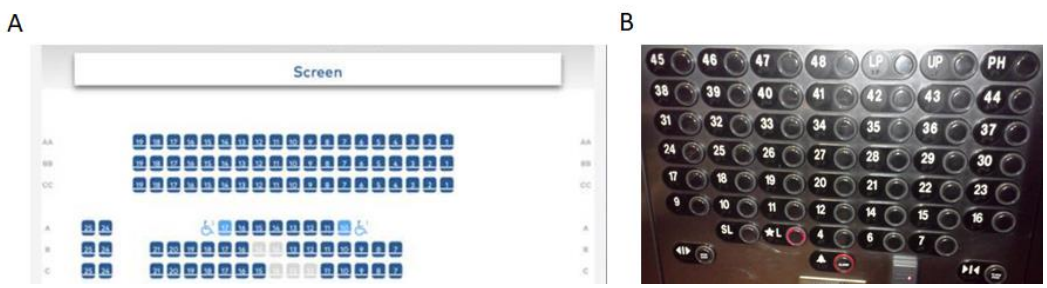

Figure 7— Cineplex seat selection map [4] (A) and elevator buttons layout (B)

The first reason that I believe that this interface has a good representation is that the relationships are explicit. Observing the layout of the seats and the order of the rows, one can easily understand the placement of the seats and their distance from the screen. The second reason is that this representation excludes extraneous details that do not affect the decision of the users in choosing their seats. These details can include the color of the seats, the shape of the foldable trays or the size of the screen.

An example of the bad interface representation is the elevator buttons layout for high-rise buildings (e.g., “Figure 7B”). The buttons provided in the interface are used for interactions. Each button is labelled separately. The floors are shown by numbers. Other graphical and abbreviation labels are included to transfer other information (e.g., close or open the door, help).

The connection between the interface and the content has several limitations. The current arrangement of the buttons is in a horizontal order and is not relevant to the order of the floors. Also, there are buttons provided and labelled in a confusing manner (e.g., “LP” or “UP”). Since these are not common abbreviations, it hinders the understanding of the meaning of the labels. The interface is cluttered by including too many labels which disperse the user’s attention. It also makes the user think more about the interface rather than the task at hand.

The first reason that I believe that this interface has a bad representation is that the relationships are not explicit. The horizontal order of the buttons is not a good representation of the floors ordered vertically. The users have to spend a great amount of time finding their desired button. The existence of the confusing abbreviations among the labels makes it even harder to find the correct button to choose. The second reason is that the current representation of the interface includes too many details. The current layout includes separate buttons and labels for each of the floor numbers. This clutters the interface. Instead of this representation, the interface could be changed to a pin pad where the specifying the floor numbers are more straight forward and has enough details (or numbers).

5 REFERENCES

[1] https://www.grainger.com/

[2] https://windowsreport.com/

[3] https://www.groovypost.com/

[4] https://www.cineplex.com/