

CS6750 HCI Summer 2021:

Assignment P3

Nan Xiao
nanx@gatech.edu

Abstract—In this assignment, first we will discuss design principles on how they help with the creation of an invisible interface and how they help to emphasize the participant view of the user. Second, an interface that is intolerant of errors are discussed, and how constraints, improved mappings and improved affordances could be used to help to avoid errors. Third, we will discuss a game and how users could make slips, mistakes and how to design to avoid that. And last, we will compare an interface with good representation and one with bad representation.

1 Q1 - DESIGN PRINCIPLES AND HEURISTICS

We will discuss how 3 of 15 design principles help to support the creation of an invisible interface, especially how they help to bridge specific phases of gulfs of execution or evaluation. Here is a recap table of Gulfs of execution and evaluation: (Joyner, 2021a)

Table 1—Gulfs of execution and evaluation

Name	Stage 1	Stage 2	Stage 3
Gulfs of execution	Identify Intentions	Identify Actions	Execute in Interface
Gulfs of evaluation	Interface Output	Interpretation	Evaluation

1.1 Principles to support the creation of an invisible interface

The three principles that can be used to support the creation of an invisible interface are: Affordances, Mapping and Consistency.

1.1.1 Affordances

“An affordance is a relationship between the properties of an object and the capabilities of the agent that determine just how the object could be possibly used” - Don Norman

The affordances can help to bridge gulfs of execution in all three stages. For

example, when we are using the chrome browser to surf the internet, we will often have multiple tabs opened. In order to close a tab, there are multiple good designs in affordances to bridge the gulfs of execution. First, the current page is highlighted, with a logo and name for the user to understand where he is. Second, a cross sign "X" is next to the page name, the user can understand that means to be used to close a tab. And last, when user hover over the mouse on the cross sign, it will change the background color to make user understand it is clickable. It also shows the full name of the page to be close to further avoid possible slips.

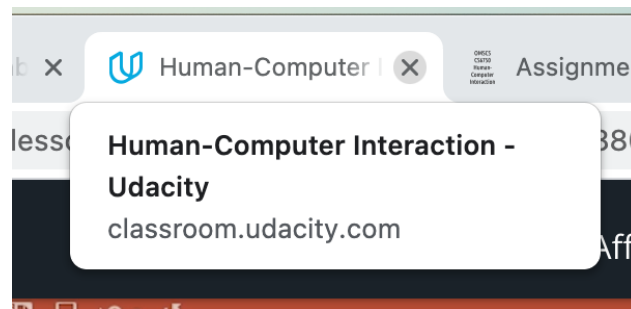


Figure 1—Close a tab in Chrome

1.1.2 Mapping

“Mapping is a technical term ... meaning the relationship between the elements of two sets of things.” - Don Norman

The mapping is a great principle to offload user’s cognitive effort. It can help to mitigate gulfs of execution, especially in what action to take and how to execute in the interface. Also, it can help to feedback in an intuitive manner that bridge the interpretation and evaluation of gulfs of evaluation. For example, when you want to add an event in the calendar app, it is easy to understand you should click the time slot you would like to add an event. And once finished, you can easily evaluate since the calendar interface will show that event in the expected time slot. The user experience is mapped between a calendar app and the actual calendar. That helps to bridge both gulfs of execution and gulfs of evaluation.

1.1.3 Consistency

“Consistency in design is virtuous. It means that lessons learned with one system transfer readily to others. ... If a new way of doing things

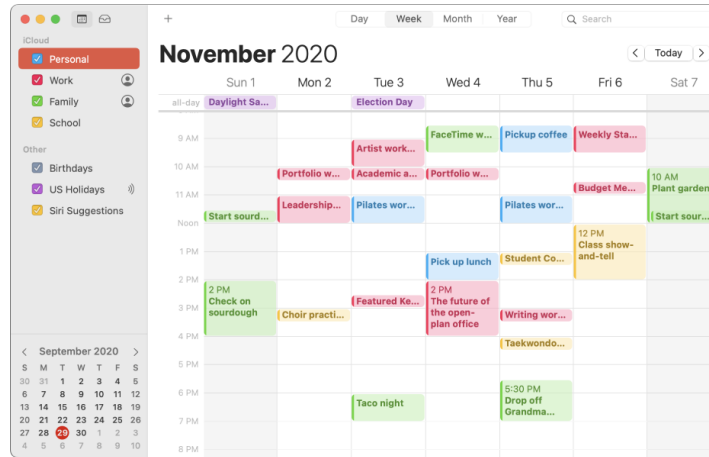


Figure 2—Calendar App

is only slightly better than the old, it is better to be consistent." - Don Norman

Consistency helps to bridge the gulfs of execution, especially on identifying actions and executing in interfaces. For example, in most of the system keyboard shortcut "CMD + W" means close the current window. So in the Chrome app, using "CMD + W" is also closing the current tab. Almost in all Mac applications, doing "CMD + W" will close the current window. This help to bridge the gulfs of execution that users do not have to learn which actions to take or how to execute to close a window. It is always the same set of commands.

1.2 Principles to create interfaces that emphasize the participant view

The two principles that can be used to create interfaces that emphasize the participant view of the user are: Equity and Tolerance.

1.2.1 Equity

"The design accommodates a wide range of individual preferences and abilities." - Ronald Mace

Equity helps to create an interface suitable for different context. For example, in the web page there is often a button to choose multi-languages. It cares about not only the user's abilities but also the language background. Also, the latest smartphones can let you adjust the brightness with your preference. So different people can use the interface in their desired brightness. Equity principle helps

to take context into consideration when designing interface.

1.2.2 Tolerance

“Users often choose system functions by mistake and will need a clearly marked ‘emergency exit’ to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.”
- Don Norman

Tolerance is important for the interface to take context into consideration. For example, it is easy to make typos when one is typing. The writing interface should be able to take care the context when user is typing wrongly and allow the user to undo. By designing a interface allowing user to make mistake, tolerance principle helps to create interfaces that emphasize the participant view.

2 Q2 - PRINCIPLES TO HELP PAYMENT CHECKOUT PAGE

2.1 Interface that is intolerant of errors

We do much more online shopping since COVID. One of the interface that is intolerant of errors the user commits is the checkout page. It is by nature that you have to key in all the information correctly, otherwise the transaction will not go through. In order to make the transaction, one need to key in all card numbers, names, expired dates, secure code correctly within a given time limit. Since it is a 16 digits code with other numerical information, it can easily goes wrong especially for the elderly. One digit wrong, normally you have to start over and key in all the information again. The transaction will only happen when you have all the numbers and information correct.

2.2 Constraints, improved mappings and improved affordances





2.2.1 Constraints


We can have the interface only allow numbers in the number box, thus will reduce the chance that people key in dashes or spaces in the card number box. Also, we can have 4 boxes with each allow only 4 digits, to further avoid the chances people key in less or more numbers.


Payment

All transactions are secure and encrypted.

Credit card





and more...





[< Return to customer information](#)

Figure 3—Checkout payment page

2.2.2 Improved Mappings

The interface can have a layout that looks like a credit card. That will map user's experience of using credit card. People will know what information to look for on their credit card. It will bridge the gulfs of execution in this case.

2.2.3 Improved Affordances

The interface can use signifier to help with perceived affordance. For example, a pop up to indicate what is secure code, or what is expired date. It will help to bridge the gulfs of execution as well, and make it easier for the user to understand what information to fill in.

3 Q3 - GAME WITH SLIPS AND MISTAKES

The game that easy to make slips or mistakes is FIFA. It is a challenging video game and people tend to make a lot of slips or mistakes in the beginning and gradually learn how to work with the Xbox controller.

3.1 Slips in the game

People tend to make slips in FIFA even for the expert player. After learned the interface for a while, people understand which buttons on the interface are mapped to which actions of the player on screen. But we can still make mistake that install of pressing B for shooting, we accidentally press X for crossing. It is a common slips people will make especially when they just start to play FIFA. It happens because there is no mapping between the buttons and the actions

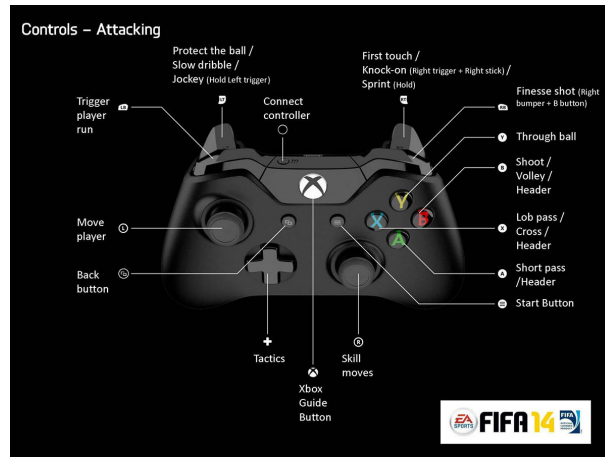


Figure 4—FIFA Xbox controller

player will take. Through practice people can gradually map B button to shooting action. But it requires cognitive effort and people may forget it from time to time and make a slip.

The interface can help to prevent that slip by adding a controller on the foot. It uses the mapping principle to help people to remember which action to take when he wants to perform a shooting action.

3.2 Mistakes in the game

Generally novice users tend to make mistakes in the game. For example, many advanced actions need users press multiple buttons to perform. Auto defence needs player holds 2 buttons together, and LB and joystick need to be press together to perform acceleration. People tend to make mistake that he do not know what to do when he want to do certain actions. It is caused by the complex interface and it is challenging for the user to remember all the button to action mappings.

The interface can have many tutorials with instructions for the user to remember how to do certain actions. To incentive the user to learn, the interface can reward the player to go through the tutorials.

3.3 Challenges in the game

It is challenging to score in the FIFA game. There is a goal keeper to stop most of the users' attempts. It is by design to make the game challenging and the users

have to learn how to shoot with best power and angle.

4 Q4 - GOOD AND BAD REPRESENTATIONS IN INTERFACES

Good representations: (Joyner, 2021c)

1. Make relationships explicit.
2. Bring objects and relationships together.
3. Excludes extraneous details.
4. Expose natural constraints.

4.1 Good representation interface

4.1.1 *Connections between the representation and the underlying content*

One interface that is using good representation is the calculator app of Mac. The interface mimic the actual calculator, so the representation of interface is directly mapped to the underlying content. User has no cognitive effort to understand how to use it if he has used an actual calculator before.

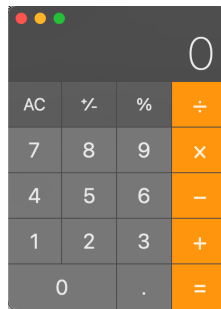


Figure 5—Mac Calculator

4.1.2 *How does the interface exemplify two criteria of a good representation*

The calculator interface makes relationship explicit. It looks like the interface of a real calculator. So it is easy for the user to understand what each button is used for. It matches the mental model exactly and offload most of the cognitive load.

The calculator interface also excludes the extraneous details. It shows the basic interface, and one can switch to the scientific calculator if needed. It does not have on/off button as the real calculator, since it is not relevant to the calculation task.

4.2 Bad representation interface

One interface that is using bad representation is the command line interface. I often need to google to understand what actions to take for the task I want to complete.

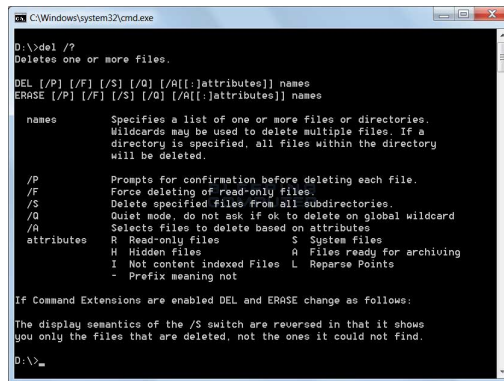


Figure 6—Command Line Interface with poor representation

4.2.1 Mismatches between the representation and the underlying content

There is no direct relationships between tasks and the commands in command line. For example, to move a file to another folder, rather than drag in the UI, you have to type "mv path/to/file path/to/destination". If one replace command "mv" to "move", the task will fail. It is not intuitive and takes much cognitive effort to remember all the commands for the tasks. And the actions to take can be difficult to remember, such as "git checkout -b test <name of remote>/test" to copy a remote branch to local for the version control purpose. There is no way to map the representation with the underlying content.

4.2.2 How does the interface violate two criteria of a good representation

The representation of this interface does not bring the objects and relationships together. It is hard to understand how to perform certain tasks. There is a slow learning curve for using command line interface to complete certain tasks.

The command line interface also violates the criteria that make the relationship explicit. There is no explicit relationship between the commands and actions. There are documentations and manuals, but the combination of commands are endless and you can complete certain tasks using multiple ways. It further increases the difficulty to be proficient in this interface.

5 REFERENCES

[1] Joyner, D. (2021a). Gulf of Execution. Udacity.

[https://classroom.udacity.com/courses/ud400/lessons/9129419879/
concepts/91335799740923](https://classroom.udacity.com/courses/ud400/lessons/9129419879/concepts/91335799740923)

[2] Joyner, D. (2021b). Gulf of Evaluation. Udacity.

[https://classroom.udacity.com/courses/ud400/lessons/9129419879/
concepts/91335799770923](https://classroom.udacity.com/courses/ud400/lessons/9129419879/concepts/91335799770923)

[3] Joyner, D. (2021c). Characteristics of Good Representations. Udacity.

[https://classroom.udacity.com/courses/ud400/lessons/9430399198/
concepts/94375211040923](https://classroom.udacity.com/courses/ud400/lessons/9430399198/concepts/94375211040923)