

# Assignment P3

Jeremy Martinez

[jmartinez91@gatech.edu](mailto:jmartinez91@gatech.edu)

## 1 QUESTION 1

### Design Principals

Discoverability	Feedback	Constraints
Mapping	Consistency	Affordances
Structure	Simplicity	Tolerance
Equity	Flexibility	Perceptibility
Ease	Comfort	Documentation

#### 1.1 Principal 1 - Discoverability

Discoverability can be used to create an invisible interface by building intuition into the interface. If a user can interact with an object that is foreign to them through an interface that is foreign to them, they must discover the actions to accomplish their goal. It must be possible to figure out what actions are possible and where and how to perform them. Further, we want to optimize our space and minimize the user's memory load by making these calls to action (CTA) visible.

***Gulf of Execution***—One can leverage known industry standards, for example, in a web app so that the user instinctively knows where the menu is, how to find more options, etc. Through A/B testing and metric analysis, you can identify what your users main goals are, and surface those calls to action quicker to narrow the **gulf of execution**. For those users trying to accomplish more technical goals, we are comfortable de-emphasizing this gulf knowing that our implementation has reduced the gulf for the majority, while still appealing to the advanced user (putting one more action in their gulf of execution).

***Gulf of Evaluation***—Although more subtle in respect to discoverability, the user must have some intuition as to how their action impacted the system. The user

must be able to discover their action had the intended impact on the object without having to dig too deep. Each action should have an equal reaction in the interface that is visible and easy to discover.

## **1.2 Principal 2 - 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," (Norman, 1988). The design affords, or hints at, the way it is supposed to be used. This creates a mapping of how a physical object should be used with its interface. Buttons are meant to be pressed, lids are meant to be lifted, wheels are meant to be rotated, etc. Just by looking at it, you can intuitively understand how it should be manipulated.

*Gulf of Execution*—This can be tricky since an affordance is "jointly determined by the qualities of the object and the ability of the agent that is interacting," (Nielsen, 1994). Judging intuition can be challenging because everyone has different experience levels and backgrounds. For example, millennials will consider an aspect of a web interface intuitive that an elderly person may not. The gulf of execution can be greatly reduced using affordances because the symbol/object suggests how the user should interact with it, provided the user recognizes the object.

*Gulf of Evaluation*—Affordances by definition should narrow the gulf of evaluation since the interface being manipulated should reflect how you expect that object to be manipulated. A button should indent, a door handle should draw toward you, lifting a lid should cause the lid to go up.

## **1.3 Principal 3 - Consistency**

Consistency should apply to one's implementation of an interface as well as the environment in which it is placed. One must stay consistent within their own interface but must also stay consistent with industry standards. "Consistency in design is virtuous," (Norman, 1988).

*Gulf of Execution*—In an attempt to reduce this gulf, one should leverage existing knowledge in their users. "Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions," (Nielsen, 1994). If consistent with the user's intuition, the interface

will feel invisible. If it is inconsistent, they will likely be confused by the interface and how to research how to use it (or lose interest and look elsewhere).

***Gulf of Evaluation***—Along the same lines as the gulf of execution, one should use industry standards for reflecting change in a UI. If a user is used to seeing a car turn right when the steering wheel turns right, then your car's wheel should do the same. A more subtle example: if you are developing a website, blue text should indicate a link to a new page, since the rest of the community has deemed this a standard. The user will evaluate their click based on if the interface has changed to a new page or not.

#### **1.4 Emphasizing the participant view**

***Principal 1 - Ease and comfort***—The ease principal states that the design can be used efficiently and comfortably and with a minimum of fatigue. The comfort principal takes into account size and space and requires that an interface be within the reach and manipulation regardless of their user's physical limitations. Both of these, by definition, take into account external factors. An application of ease of use could be how easy it is to use navigation app considering the user is driving. An application of the comfort principal would be how/where to place button on a smartphone so that it is accessible to the way most people hold a smartphone.

***Principal 2 - Flexibility***—A user interface is flexible if it adheres to different users and situations. Providing multiple options for accomplishing the same task takes into account the knowledge of the user. However, it can also take into account their environment when performing the task. The old iPod shuffle's and nano's addressed this with selecting a next song. They discovered many of their users use the mp3 player while exercising. So they introduced a feature, targeted for that context, where the user can simply shake the device left-to-right and it will shuffle or select the next song.

## **2 QUESTION 2**

### **2.1 Interface from everyday life - Venmo**

[Venmo](#) is a mobile application for sending/request money. Venmo is intolerant to errors, however, this is also a fundamental feature of the application. The interface has only five calls to action (CTA). Write a new request/payment,

view payments of everyone/friends/personal, and the menu. When selecting a payment, you have to click the button twice, however, once you send a payment there is no undoing it. If this was an error, incorrect amount, wrong person, etc. there is no way to correct the mistake. This is partly what makes Venmo an acceptable payment method for transaction like in Craigslist, because the seller has certainty that the payment cannot be revoked. However, it is not a tolerant UI for mistakes (see Appendix A for screenshots).

One **constraint** that could be put in place is when sending money to an account for the first time. This could potentially mean there was a typo when entering the username and an extra screen asking the user if they know the recipient and intend to actually send it to this account would help deter people from making this mistake. This constraint would force the user to think twice and click twice that they do in fact want to send money to this completely new account they have never interacted with before.

Improved **mappings** could be put in place here to show the user exactly who they are sending money to. If the user does not have a profile picture with their account, the UI should provide a link to their corresponding Facebook account (if applicable). That way, the user can match the profile picture with the face of the person they want to send money to (if they have met face-to-face). Often times, just the user's initials appear with a username, which is sometimes cryptic and not indicative of the actual person you intend to send money to.

One **affordance** I've always enjoyed on mobile payment platforms is Amazon's. If you have a payment method on file, and your cart has an item in it, when clicking buy now you are prompted with a slide to complete the transaction. When sliding this bar (left to right), it feels like you are sending your money on over with the transaction. This affordance may be a bit of a stretch, but the action must be intentional, and one could argue it mimic's the act of handing over cash in a hand-to-hand transaction. If Venmo applied this same affordance to their UI, it would make the transaction feel more concrete (see Appendix B for screenshots).

### 3 QUESTION 3 - SUPER SMASH BROS

A slip that sometimes occurs in the video game Super Smash Bros for Nintendo 64 is dying when trying to jump back to the fighting platform. The specific

character in focus for this is Jigglypuff. This slip sometimes happens with this character because it is the only character without a secondary jump. For every other character, the button combo UP-B will cause a secondary jump. For Jigglypuff, this button combo is changed to sing. The interface could be changed instead so that Jigglypuff's attacks are consistent with other characters.

A mistake that sometimes occurs in the video game is falling off the map (and dying) after picking up the hammer. This mistake happens because the behavior/physics of the character changes when the hammer is picked up. And so, normal survival skills that the user is accustomed to, like the ability to grab onto the ledge, are unavailable when they have wielded the hammer. One way they could change this is to allow certain survival basics while wielded with the hammer like all other items.

An aspect of the game that makes it difficult, but not necessarily causing a mistake/slip are different elements of a map that inflicts damage. For example, on the Star Fox map, at random, spaceships shoot lasers at the players inflicting damage.

#### 4 QUESTION 4

Choosing an interface from my everyday life, the tool observed is [Sublime Text 3](#). Sublime text is a text editor with added features for syntax highlighting and plugin support like git, linting, type checking, etc. The connections between the files in the editor directly map to the file system, and this is intuitive through the UI.

- Analogy - The Sublime Text editor is analogous to the file system because it reflects the directory structure identically.
- Learning Curve - The Sublime Text editor has a really low learning curve for performing basic function like editing, saving, deleting files, etc. However more intricate tasks like plugin installation and setup add to the learning curve. However, to remedy this, the editor prompts you with external links on exactly how to perform these tasks.

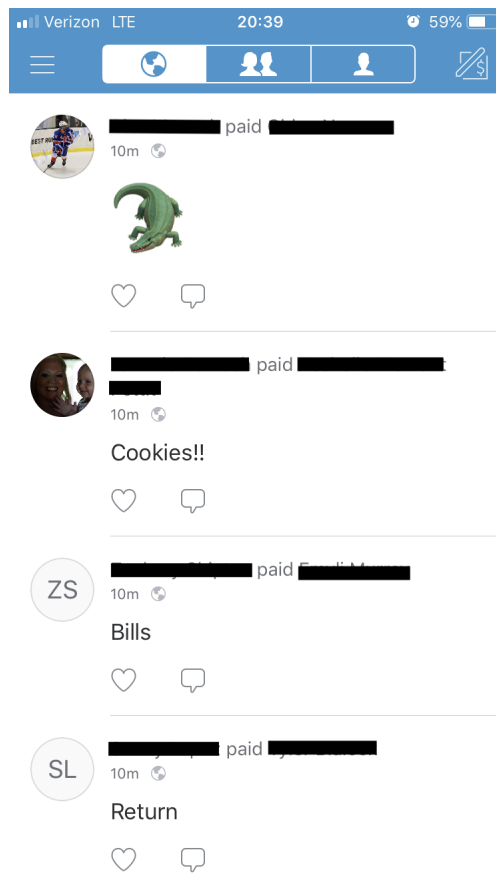
An application that attempts to accomplish the same thing is [Vim](#). Vim is an editor within the terminal. The reason vim has a less intuitive implementation is because navigating the entire app is operated solely through the keyboard without the mouse. This is inconsistent with other applications in the OS making

the learning curve much, much higher. However, it is analogous to the rest of the terminal in respect to the file system. The representations of the filesystem is also worse because you cannot see the entire directory structure in one view, you must enter into a directory where you can only see the top layer of the directory.

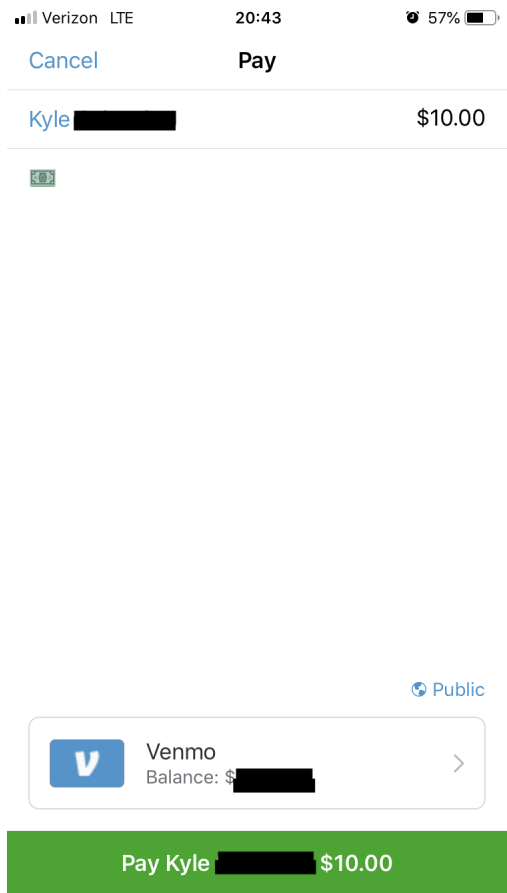
## 5 REFERENCES

1. Norman, Donald (1988). "Preface to the 2002 Edition". The Design of Everyday Things. New York: Basic Books. ISBN 978-0-465-06710-7.
2. Nielsen, Jakob. Usability Inspection Methods. New York, NY: John Wiley and Sons, 1994.

## 6 APPENDIX A - VENMO SCREENSHOTS



**Figure 1.** Venmo home screen.



**Figure 2.** Venmo payment screen.

## 7 APPENDIX A - AMAZON SCREENSHOTS

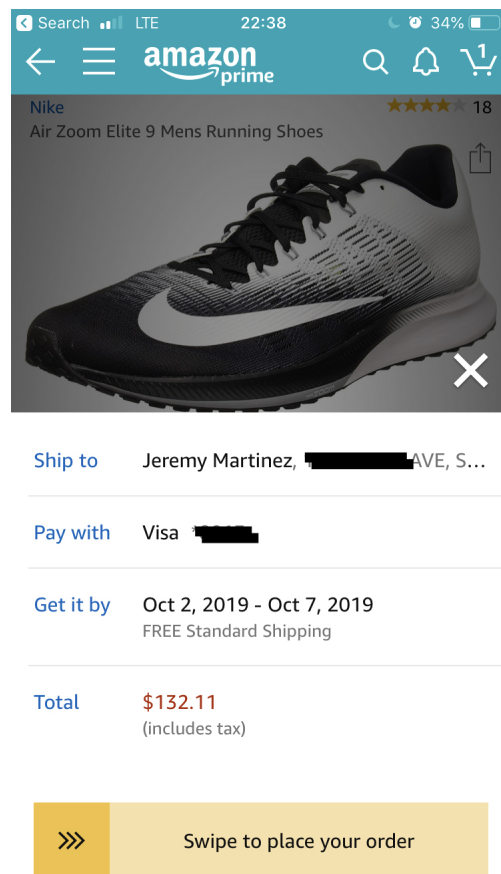


Figure 3. Amazon buy-now screen.