

Evaluating Human-Computer Interaction Theories through Practical Experiments

Bachelor Thesis
Faculty of Science, University of Bern

submitted by
Julien Gaumez
from Sigriswil, Schweiz

PD Dr. Kaspar Riesen
Institute of Computer Science (INF)
University of Bern, Switzerland

Abstract

This thesis takes a second look at existing laws, rules and principles from the field of Human Computer Interaction. Six theories are picked and reevaluated through practical experiments. The selected topics are Fitts' Law, Shneiderman's Ressource Model, Recognition vs Recall, Digital Affordance, Dialog Feedback and Mode Errors. The used method is to create an interface in which the theory gets applied and directly comparing it with a version of the same interface that does not apply the theory. The comparison is done by collecting data on user efficiency and experience on both interfaces. The results show that most theories lead to drastic improvements, but one theory is seemingly outdated. Also the application of the theories is not always straight forward. The general conclusion is that theories from the field of Human Computer Interaction definitely help in creating better usability. The weakness of the theories comes in their simplification and generalization. Importantly with the continuously evolving nature of both the human and the computer sides of human computer interactions, theories need to be kept up to date and adapted to new technologies.

Acknowledgements

I would like to express my sincere gratitude to my supervisor PD Dr. Kaspar Riesen for his guidance and support throughout this project. I would also like to thank anyone who participated in one of the experiments and provided valuable feedback. Their contributions were essential in making this research possible. Lastly I want to give a special thanks to my study friends who gave me their opinion on key decisions on the experiment setups and in the writing process of this thesis.

Contents

1	Introduction	1
1.1	Human-Computer Interaction	1
1.1.1	Definition	1
1.1.2	Subcategories and the Experiments	1
1.2	Research Objective	2
1.3	Structure of Thesis	3
2	Fitts' Law	4
2.1	Underlying Concepts	4
2.1.1	Placement in Human-Computer Interaction	4
2.1.2	What is Fitts' Law	4
2.2	Experiment Setup	5
2.3	Results	6
2.4	Discussion and Interpretation	8
3	Shneiderman's Resource Model	9
3.1	Underlying Concepts	9
3.1.1	Placement in Human-Computer Interaction	9
3.1.2	What is Shneiderman's Resource Model	9
3.2	Experiment Setup	10
3.3	Results	11
3.4	Discussion and Interpretation	12
4	Recognition vs Recall	13
4.1	Underlying Concepts	13
4.1.1	Placement in Human-Computer Interaction	13
4.1.2	Atkinson–Shiffrin memory model	13
4.2	Experiment Setup	14
4.3	Results	15
4.4	Discussion and Interpretation	16

5	Digital Affordance	18
5.1	Underlying Concepts	18
5.1.1	Placement in Human-Computer Interaction	18
5.1.2	Gulfs of Execution and Evaluation	18
5.1.3	Affordance	19
5.2	Experiment Setup	19
5.3	Results	21
5.4	Discussion and Interpretation	22
6	Dialog Feedback	23
6.1	Underlying Concepts	23
6.1.1	Placement in Human-Computer Interaction	23
6.1.2	Rules for Feedback	23
6.1.3	Two types of Feedback	24
6.2	Experiment Setup	24
6.3	Results	25
6.4	Discussion and Interpretation	26
7	Mode Errors	27
7.1	Underlying Concepts	27
7.1.1	Placement in Human-Computer Interaction	27
7.1.2	Error Types	27
7.1.3	Mode Errors	27
7.2	Experiment Setup	28
7.3	Results	29
7.4	Discussion and Interpretation	30
8	Conclusions and Future Work	31
8.1	Overall Conclusion	31
8.2	Future Work	32
A	Appendix Title	34
A.1	More Plots to the Experiment on Fitts' Law	34
A.2	Websites in Affordance and Flat Design	37
	Bibliography	41

Chapter 1

Introduction

1.1 Human-Computer Interaction

1.1.1 Definition

There are multiple definitions for the field of Human-Computer Interaction (HCI) that vary in the range of topics they include. The Association for Computing Machinery uses the following one [1]:

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

In other words, HCI is concerned with understanding how humans interact with computers, to then apply this knowledge in the design process. This way we can create technologies that allow for a natural and intuitive user experience. The field has its roots in many different research fields that go far beyond computer science. The most important ones are among the following: Computer graphics, operating systems, human factors, ergonomics, industrial engineering, cognitive psychology, and the systems part of computer science. Overall anything that influences the way humans interact with computers is a part of HCI.

1.1.2 Subcategories and the Experiments

Because of the wide range of topics in HCI, one can define many sub-fields to divide HCI into. Most of these sub-fields take knowledge from another research field and apply it in a HCI context and thus are named after the other research field. In this thesis I conduct six experiments that can be categorized with the following sub-fields:

- *Human factors and ergonomics*: This sub-field is about the characteristics of people and how they influence their interactions with products and systems.
- *Cognitive ergonomics*: This sub-field focuses on the study of human cognition and how technologies should adapt to it.
- *Feedback*: In any human-computer interaction the human needs to get feedback, this sub-field studies everything that needs to be considered to create good feedback.
- *User interface design*: This sub-field focuses on all tips and tricks designers should apply in their interfaces.
- *User experience design*: Here the focus is on improving the experience of the users when using the product.

The first three experiments focus on human factors and ergonomics, with the second and third one specifically being about human cognition. The first experiment is about button placement to improve efficiency. The second one focuses on the limited cognitive resources available to users. The third one is about how human memory works and how computer systems should adapt to it.

The second half of the experiments are mainly about creating better user experiences by improving visuals and feedback of interfaces. The fourth experiment compares two design styles and whether one is more intuitive and thus easier to use than the other. The fifth and sixth experiment compare different types of feedback and their influence on efficiency and errors respectively.

All six experiments are connected to the sub-fields user interface design and user experience design in some way. These two sub-fields are connected to every human-computer interaction and thus overlap with the other sub-fields.

1.2 Research Objective

As we just defined, the field of HCI is concerned with understanding how humans interact with computers and applying this knowledge to design technologies that allow for efficient but still natural and intuitive user experiences. Despite the significant progress that has been made in this field, there is still much to be learned and improved about the complex relationship between humans and computers.

In this thesis, I aim to contribute to the field of HCI by conducting a series of experiments specifically designed to test and reevaluate existing theories. The research question is: “How well do the existing laws, rules and principles of HCI align with the reality of human-computer interactions, and where do these theories need to be further improved?” Through the experiments, I hope to confirm or challenge the existing laws, rules or principles and gain new insights into the field.

Another motivation for this thesis is to create experiments that show the impact rather small design decisions can have on the usability of the final product. The experiments should show the importance of HCI to new developers. To do this the experiments need to stay simple and generic.

1.3 Structure of Thesis

Chapters 2 to 7 each discuss one of the experiments I conducted. They are in the same order as they were introduced in Section 1.1.2. All the chapters about an experiment are structured the same way. First the underlying concepts are explained. This includes their placement in HCI and the theory the experiment will test and reevaluate. The second section describes the setup of the experiment in detail. This is followed by the results of the experiment and lastly the discussion and interpretation of these results.

Once the six experiments were discussed in their own chapters, there is a last chapter that contains a final conclusion and an outlook for future research. In the appendix, after all major chapters, some extra bits and pieces from specified experiments or their results can be found.

Chapter 2

Fitts' Law

2.1 Underlying Concepts

2.1.1 Placement in Human-Computer Interaction

The focus of this experiment is on the human element of HCI, specifically the human limitations of physical movement. In order to increase efficiency in human-computer interfaces we need to understand what humans are capable of and how they work. We can then use this knowledge in the design process of interfaces. The goal of this experiment is to analyse, what effects the time a user requires to click a button. Such an interaction is used in most interfaces and thus is relevant for most human-computer interactions. The foundation for this experiment gives us Fitts' Law that emerged from the field of experimental psychology.

2.1.2 What is Fitts' Law

Fitts' Law is a predictive model of human movement and gives us a formula to compute the movement time required to point at a target. The formula is expressed as:

$$MT = a + b \log_2 \left(\frac{2D}{W} \right)$$

where MT is the movement time, a and b are empirically determined constants, D is the movement distance and W the width of the target [2].

This is the original formula introduced by Fitts. Many researchers have tried to improve the model, which lead to multiple slight variations of the formula. Although the variations change some assumptions in edge cases, they can all be interpreted the same way. The movement time decreases with shorter distance to the target and the bigger the target. This relation between distance and width is often referred to



Figure 2.1: Examples of the infinite target size.

as the index of difficulty.

$$ID = \log_2 \left(\frac{2D}{W} \right)$$

Two Components of Movement

Using the formula one can see that doubling the distance or the size of a target does not result in a doubled movement time. This is the case because the pointing movement can be split into two components. First a fast and imprecise movement to get near the target. This is then followed by a slower and accurate movement to ensure hitting the target [3].

Infinite Target Size

In the specific case of clicking buttons on a screen using a cursor, designers can make use of the screen's edges. As the cursor is implemented such that it cannot leave the screen, targets that go to the edge of the screen have an infinitely big size. Which lowers the movement time according to Fitts' Law.

An example of this is the location of the close button of any program running on a Windows computer shown in (a) of Figure 2.1. It is always located in the top right corner, all the way on the edge of the screen. Thus a user can drag the cursor to the top right at full speed and does not need to worry about accuracy as overshooting is impossible. An example where this is not used is shown in (b) of Figure 2.1. Windows 11 introduced tabs to the explorer, they look and behave the same as tabs users are already familiar with from browsers. But for some reason the button to swap to a different tab stops a few pixels before the edge of the screen. Thus not only losing the benefit of the infinite target size but even breaking the consistency of other applications with tabs.

2.2 Experiment Setup

With this experiment the goal is to show where buttons should be placed in order to maximize the users efficiency. In particular in which cases the benefits of placing them on the edge of the screen, using the infinite target size, outweigh the

longer distance caused by placing them on the edge. We will also see if the general interpretation from Fitts' Law holds up.

To create such an experiment I decided to write a python program that shows a red target on an otherwise white screen. Once this target is clicked the next target appears. This process is repeated until 30 targets have been clicked. To compare the results there are four different types of targets with each of them getting their own 30 target version of the experiment. The types of targets are small targets on the edge, small targets not on the edge and both of them with big targets respectively. Participants of my experiment complete all four versions. This means every participant hits a total of 120 targets.

To minimize the impact of chance, the 30 small and big targets on the edge have the same center and the big targets have twice the width and height. Same applies to the small and big targets not on the edge. This makes it easy to compare the versions with small and big targets afterwards. The downside is that participants might have an advantage in the second run if they remember the sequence of the targets. To reduce this downside, the order in which participants complete the different versions is decided randomly and the participants are not told that the small and big versions share the same sequence of target locations.

The second parameter in this experiment is the input device. In my experiment I collect data on three input devices. The touch-pad of a laptop, a computer mouse and a stylus that can be used to interact on the touch screen directly. I chose the stylus over the use of fingers because it allows for more precision and makes sure participants do not use their second hand to avoid long distance movements. For touch-pad and mouse participants are allowed to change the sensitivity settings to match their preference.

2.3 Results

Figure 2.2 shows six scatter plots of the collected data. Each plot compares the small version with the identical big version of the experiment. The straight lines on the plots are regression lines computed on the small and big data-points respectively.

The results show that the bigger targets require less time to be hit than their smaller counterparts. As shown in Table 2.1 this is true for every input device

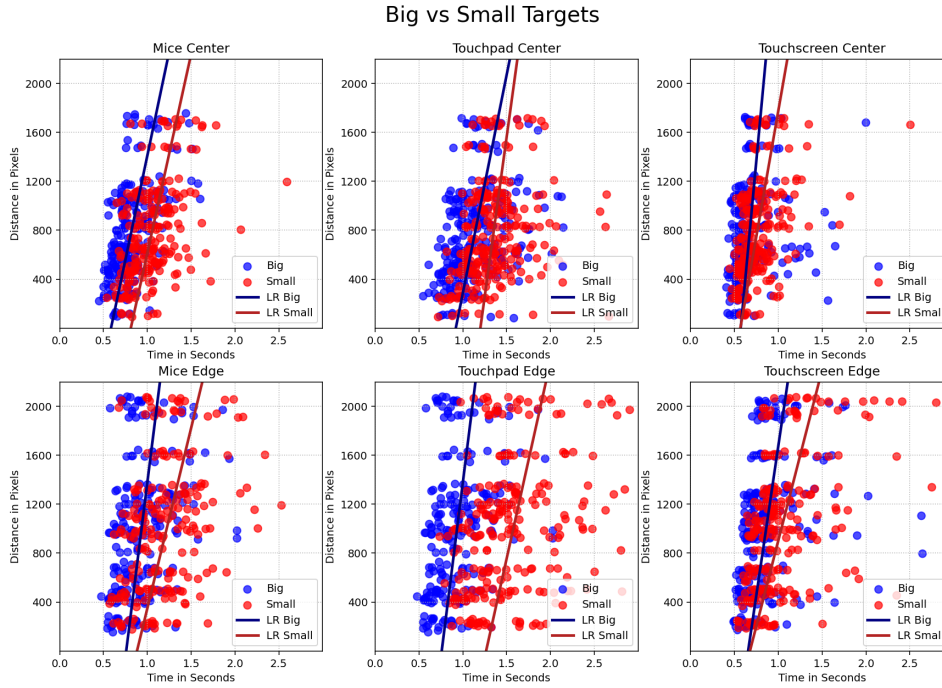


Figure 2.2: Comparison between big and small targets.

Version	avg Time Small	avg Time Big	difference	p-value
Mouse Center	0.82	1.05	0.23	0.000
Touch-pad Center	1.14	1.35	0.21	0.000
Touch-screen Center	0.67	0.76	0.08	0.000
Mouse Edge	0.94	1.23	0.29	0.000
Touch-pad Edge	0.94	1.58	0.65	0.000
Touch-screen Edge	0.86	1.04	0.18	0.000

Table 2.1: Comparison between average time to hit small and big target.

and no matter where the targets are located. With seven participants using the mouse and eight participants for both other input devices, each hitting a total of 120 targets, the results contain a statistically significant difference. The respective p-values of the t-test are shown in the table.

My collected data does not show a benefit to placing targets on the edge of the screen, contrary the participants were slower to hit targets on the edge compared to similar targets not on the edge. The amount of clicks off target are significantly increased by smaller target size for targets located on the edge. Targets located in the center only show such a significant difference with the stylus as the input device. More plots and tables to this part of the evaluation can be found in Section A.1 of the appendix.

2.4 Discussion and Interpretation

The expected influence of the targets size and location on the movement time match the results of my experiment. A bigger and closer a target is the smaller the time required to hit it. This is true for all three input devices. The targets on the screens edge did not show an improvement compared to targets somewhere else on the screen.

I believe this is the case because in my experiment the time the participants need to spot the new target is included in the movement time. Targets on the edge likely take more time to spot than targets in the center of the screen. This would also explain why the times for the big targets are quite similar, no matter the location, but for the small targets the difference in movement time is much bigger depending on the location. An observation that shows that the placement of buttons on the edge of the screen can be beneficial, is that if we compare the best times, instead of the average, of specific targets and distances the targets on the edge win for mouse and touch-pad.

In conclusion I suggest to use the edges of the screen and especially the corners in cases where the user already knows the buttons location. This is the case for interfaces the user uses regularly or buttons with consistent function and placement across multiple programs, like the close button on Windows. In cases where most users will be new to the interface, placing buttons in a more visible spot near the center of the screen should prove to be the better option. What never should be done are buttons that stop pixels before the edge of the screen, such as the tabs in the windows explorer. Buttons next to an otherwise empty edge of the screen should always go all the way to this edge.

Chapter 3

Shneiderman's Resource Model

3.1 Underlying Concepts

3.1.1 Placement in Human-Computer Interaction

For this experiment we stay on the topic of human limitations and how we should adapt to them. In this chapter we dive into the human process of speech recognition and its cognitive load. Research shows that speech based interfaces bring the downside of interfering with the short-term and working memory used for problem solving and recall.

3.1.2 What is Shneiderman's Resource Model

Ben Shneiderman is the first to talk about this problem and what it means for HCI in his article "The limits of speech recognition" [4]. He presents his cognitive resource model shown in Figure 3.1. It shows that the cognitive resources used for problem solving and recall overlap with those used for speech in- and output while the ones used for hand-eye coordination are separate. He even argues that these resources expand while pointing and clicking.

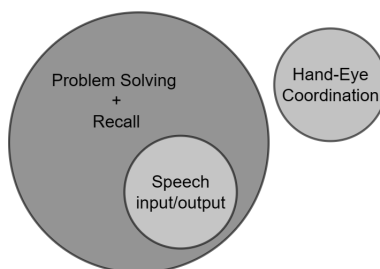


Figure 3.1: Shneiderman's cognitive resource model.

While working on tasks with a high cognitive load, people using speech to interact with a computer take much longer and sometimes are more error prone than their counterparts working with keyboard and mouse. Multiple researchers stumbled upon this phenomenon while analyzing their work process. Interface designers need to be aware of this to avoid overloading the cognitive resources available to their users.

Today there are studies centered around this human limitation. For example the study from 2020 "Effect of Speech Recognition on Problem Solving and Recall in Consumer Digital Health Tasks: Controlled Laboratory Experiment" [5] has the objective to evaluate if the phenomenon appears in some specific consumer digital health tasks. The conclusion is that keyboard and mouse are preferable to speech recognition for complex tasks involving problem solving and recall.

Most of the research compares users speaking to give commands with users using keyboard and mouse to do so. Shneiderman's resource model says that the same phenomenon occurs when the user needs to listen to something. That is what my experiment will be testing.

3.2 Experiment Setup

This experiment has the objective to evaluate the influence of different tasks on the human recall capability. Does listening and processing the heard information significantly limit ones ability to remember things. Does hand-eye coordination truly have zero negative effect or even a positive one on recall, as stated by Shneiderman?

During the experiment participants are shown ten images, each one for six seconds. Every image shows an easily recognizable animal, object or landscape. Participants are tasked to recall as many images as they can after all ten images have been shown. To do this they get a text input field, in which they describe each image they remember with a single word. This basic memory recall task is then combined with a distraction to create one version of the experiment. In total there are three versions including a control version without any distraction. One distraction is based on hand-eye coordination and the other one on listening. Participants do all three versions in a random order.

The hand-eye coordination based distraction is similar to the experiment on Fitts' Law described in Chapter 2. Big red targets appear on top of the images. Clicking a target with the cursor creates the next randomly generated target. There is always exactly one target at a time. If no target has been clicked for two seconds the current image is removed, leaving a black screen with only the red target on it. Once the target gets clicked the image comes back. The time of the images keep ticking down during the black screen. This forces the participants to keep working on the distraction task throughout the 60 seconds.

For the listening distraction participants hear simple math calculations and are tasked to say something whenever a calculation is wrong. The calculations consist of natural numbers in one of the two following forms: $a + b = c$ or $a - b = c$ with $0 \leq a, b \leq 10$. On average 30% of the calculations are wrong, in which case c is chosen randomly between 0 and 20. The decision for this as the listening distraction instead of a story or other text is that I can check if they are actively listening, during the experiment and not only after. Which would force the participants to remember not only the images but also details of the text they heard. The downside of this decision is that participants need to be able to do some mental math. I take this trade off because most of my participants are math or computer science students and should not have any problem with such calculations.

3.3 Results

In Figure 3.2 the collected data of the experiment is shown. Plot (a) on the left shows the distribution of how many images the participants recalled correctly, the achieved score, in the respective version of the experiment. Plot (b) on the right shows the amount of incorrect images the participants mentioned. Incorrect mentions are mentions where the participants recalled an image that was not a part of the ten images that were shown to them. The number of images the participants forgot is ten minus the amount of correct mentions.

The clicking version was completed by nine participants, both other versions by eight. The control version, the one without distraction, reached the best average score with 8.5 correctly recalled images out of 10. Adding the clicking distraction dropped the average score to 7.4. The version with the listening distraction has an average score of 4.1, which is less than half of what was achieved in the control version.

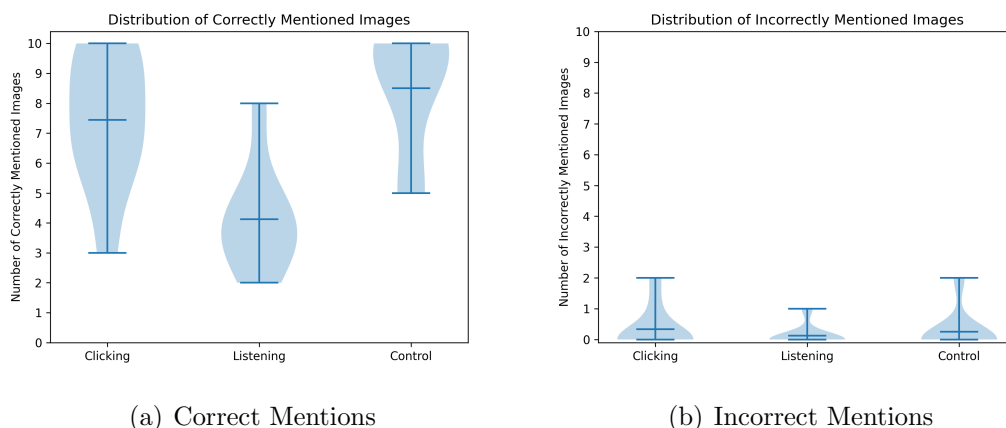


Figure 3.2: Distribution of the results.

By using Tukey HSD to make a pairwise comparison of the achieved scores we get the following statistical evaluation. The difference between the control and clicking version leads to a p-value of 0.570 and thus is not statistically significant. The comparison of the listening version with the control and clicking one gives a p-value of 0.001 and 0.011 respectively. These p-values show that the difference between these versions is statistically significant. Comparing the amount of incorrect mentions in the same way gives us p-values of 0.958, 0.769 and 0.914 meaning there is no statistically significant difference.

3.4 Discussion and Interpretation

The results of my experiment match the expectations from the theory. The participants had a hard time memorizing and then recalling the images while listening to the math calculations. Although the difference is statistically significant and the results match the theory they do not prove it. As the worse performance in this version could also be caused by the mental math and not the listening itself.

While in my test group the difference in the score of the clicking distraction and the control version is not statistically significant, on average the participants recalled one image less. Thus the argued expansion of the cognitive resources for recall while pointing and clicking does not occur in my case.

The incorrect mentions do not seem to be strongly influenced by the distractions. Most of them were caused by doing multiple versions of the experiment directly after each other. But there were also two mentions of images that do not exist in the data-set of images used in the experiment.

Chapter 4

Recognition vs Recall

4.1 Underlying Concepts

4.1.1 Placement in Human-Computer Interaction

In this chapter we stay on the topic of human memory, more precisely the process of retrieving information from our memory. This knowledge is necessary for interface designers to provide a good user experience. Computers often require to ask the user for some information. In such an interaction the human is a limiting factor once again. Interfaces should use the user's easiest possible way of giving the necessary information. To accomplish this, first we need to take a look at how the human memory works.

4.1.2 Atkinson–Shiffrin memory model

In 1968 Atkinson and Shiffrin present a memory model (Figure 4.1) that defines three memory functions [6]. Sensory memory (SM), short-term memory (STM) and long-term memory (LTM). Sensory memory holds all the input we get through our senses. In the STM we process information and can recall it within about a minute. The amount of information we can keep in our STM is very small, generally about seven chunks of information. When we need to store large amounts of information or keep it for a long period of time, the LTM is used. A limit for its capacity has not been found yet. We can also remember things for a lifetime, meaning there is



Figure 4.1: Atkinson and Shiffrin's memory model.

neither a relevant limit to the duration of how long we can store information.

Working Memory

Baddeley and other researchers show that Atkinson's and Shiffrin's memory model is too simplistic. They replace the concept of STM with working memory (WM). Baddeley describes WM as a complex interactive system that is able to provide an interface between cognition and action [7]. He keeps LTM as a component of his model that interacts with the WM.

Retrieving Information

To access the information stored in our LTM we need to retrieve it. Researchers distinguish between two types of retrieval from LTM. Retrieving details stored in our LTM is known as recall. The second type is recognition, that is what happens when we are familiar with something. Recognition feels easier, as we have more information about what we are looking for in our LTM. Kayacan, Atesgoz and Sak show this in their study that compares recall and recognition [8]. Participants using recognition achieve higher scores than the participants forced to use recall. With their experiment setup participants were tested on information, they learned minutes before. This is where my experiment is different, participants of my experiment are tested on memory they acquired over years.

4.2 Experiment Setup

The purpose of this experiment is to compare the use of recall versus the use of recognition in interfaces. What is the best way to ask a user for information he has stored in his LTM? Can we provide a good user experience while getting a lot of information or do these two objectives interfere with each other?

For this experiment I chose a scenario in which the computer wants information about the users preference in TV shows. To get this knowledge the user makes a list of TV shows he likes and TV shows he dislikes. This is information participants stored in their LTM a long time ago. To test recall versus recognition I create and compare two interfaces that each require one of the two types of memory retrieval. You can see the final versions of the two interfaces in Figure 4.2.

Recall interface The recall interface makes use of text input. The users get all instructions to their task and are then presented two labeled text input fields and

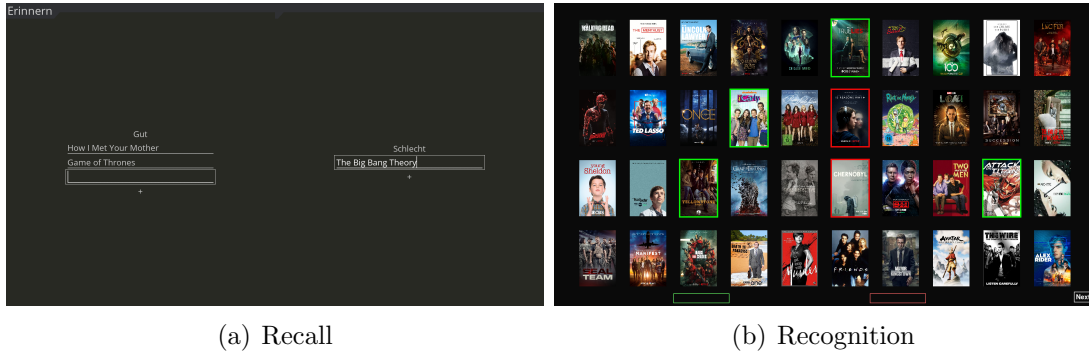


Figure 4.2: Examples for both interfaces

two buttons below them. On the left side they write the names of TV shows they like and on the right the ones they dislike. With the buttons below the input fields they can generate as many input fields as they need.

Recognition interface The second interface shows multiple TV show posters to the user. Participants can mark TV shows as good with one click on the poster. A second click marks it as bad and by clicking it a third time the TV show is set back to neutral. The current mark of a TV show is displayed with a green or red border around the poster. In total there are 120 TV shows, ready to be marked, split onto three pages. In case a TV show the user likes or dislikes, that is not shown, pops into his mind, there are text input fields similar to the recall interface at the bottom of the pages.

In the experiment participants get 90 seconds to create their lists of good and bad TV shows with one of the two interfaces. After the 90 seconds the interface stops taking new inputs. After checking that all text inputs truly are TV shows, the amount of good and bad mentions is stored. Participants are then asked to rate two aspects of their experience with the interface from one to five. How tedious was it to fulfill the given task and how happy are you with the selection of your TV shows.

4.3 Results

Both versions of the experiment were conducted with seven participants. On average 18.0 TV shows were listed with the recognition interface, with 14.9 being the average amount classed as good and 3.1 as bad respectively. The text input boxes on the recognition interface were not used a single time. On the recall interface the average of total TV shows is 9.0, with on average 6.9 on the good and 2.1 on the

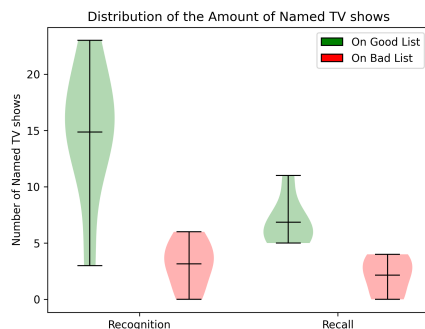


Figure 4.3: Distribution of number of TV shows on each list.

bad list. Figure 4.3 shows the distribution of the different lists. The distributions with the green background represent the good lists and the red ones the bad lists.

Comparing the total amount of TV shows with the Mann-Whitney-U-Test conducts a p-value of 0.029. Comparing only the amount of TV shows on the good lists also results in a p-value of 0.029. The p-value of the comparison of TV shows on the bad lists is 0.364.

The task got rated more tedious on the recall interface than the recognition interface, scoring an average of 3.0 and 1.7 out of five respectively. Mann-Whitney-U-Test gives a p-value of 0.034 on the comparison of these ratings. The second rating, how happy the participants are with their selection, is also in favor of the recognition interface. It scored an average of 3.9 and the recall interface scored an average of 2.7, resulting in a p-value of 0.085.

4.4 Discussion and Interpretation

The results show clearly that an interface based on recognition is more user friendly, when asking humans for information. The process is not only less tedious, the user will also be able to give more and better information especially when it is about examples.

I think the key thing about recognition based interfaces are the examples. In my case the spectrum of TV shows is big and my examples consist of the shows that I know of and the top 100 most popular shows. This set of TV shows worked great for people my age. The one participant that is not in this same age group only marked four TV shows in the recognition interface. Which is not caused by the

participant only knowing four TV shows, as they listed 12 TV shows in the recall interface.

The added possibility of naming things by recall in the recognition interface was not used by my participants. It is tough to say if this is specific to my experiment or something that happens in general, when adding this option to an otherwise recognition based interface. In my implementation, this option was at the very bottom. Thus participants might just have forgotten about it, but I think this is not the only reason. I believe that the participants ignored the text input as it is more tedious to recall TV shows than to mark examples. Thus even after only recognizing one out of 20 TV shows, they would rather keep on looking through the examples than switching to the text input and recall.

In conclusion I would say, it is probably better to put text input fields after the user scanned through all examples, than to have this option somewhere out of focus all the time. To create a better list of examples, make your interface adapt the examples it gives the user according to their previous answers. For example after the first ten TV shows, if the user marked a TV show of a specific genre show him more TV shows of this genre.

Chapter 5

Digital Affordance

5.1 Underlying Concepts

5.1.1 Placement in Human-Computer Interaction

Whenever a human interacts with a computer, they have a goal they want to achieve. To accomplish this they define a bunch of steps that they need to take. They then proceed to interact with the computer, executing one step after the other. At the end they check if everything worked as planned and whether or not they accomplished their goal.

This interaction process lies at the bottom of every human-computer interaction and is the topic of this chapter. Donald Norman presents two gulfs that separate the mental state of the human and the physical state of the computer in his book "The Design of Everyday Things" [9]. The gulf of execution and the gulf of evaluation. These concepts prove very helpful to understand the difficulties that come up in human-computer interactions.

5.1.2 Gulfs of Execution and Evaluation

Execution Norman describes the gulf of execution as the difference between the intentions and the allowable actions. This includes all the things the user planned to do that the computer does not allow them to do. The gulf of execution is a measurement of how close the actions provided by the system match the users intended course of actions.

Evaluation The gulf of evaluation gives a measurement of how difficult it is to keep track of the current state of the system. The user should be able to evaluate

what happened and what is about to happen with as little effort as possible. This allows the user to know whether they can still follow their initial plan or if they need to adapt to something.

A good computer system keeps both gulfs small and builds bridges to overcome them. There are five fundamental design principles that can be used to do so. Affordance, Signifier, Constraints, Mapping and Feedback. We will now take a closer look at Affordance.

5.1.3 Affordance

The term affordance describes all actions a human perceives they can do given a specific object. It is a form of intuition on how to handle and what to do with an object. Any object with a handle has the affordance to be pulled or picked up. The affordance of a tool or other device helps new users understand them. It limits the possible actions a user might plan before interacting with them.

We want to keep this form of intuition in our digital interfaces. This proves tricky, as the shape and look of a digital element has no direct influence on the capability of its actions. Take a button in an interface for example. Its color, size and position can be set arbitrarily by the designer, no matter what the button does. This is different from real world objects, which have to follow the rules of physics.

To compensate for the lack of natural affordance, designs often stay realistic and make use of skeuomorphism. A skeuomorphism is design that keeps the structure or appearance of older, familiar things. Designing a print button to display an icon that represents a physical printer is an example of a skeuomorphism. Another way to add affordance is to add a shadow around the button. This makes the button look three dimensional and thus indicates that it can be pushed. Chen, Fadel and Mata show that such design decisions can improve user efficiency in user interfaces [10].

5.2 Experiment Setup

This experiment will check what influence adding affordance, like adding shadows, has on the users interaction with the system. Is Flat Design, a new design trend that eliminates affordance for the sake of clean aesthetics, a backwards step in usability?

To determine this I select five different website templates with flat design. I then add affordance to all interactive elements of the templates to create a second version

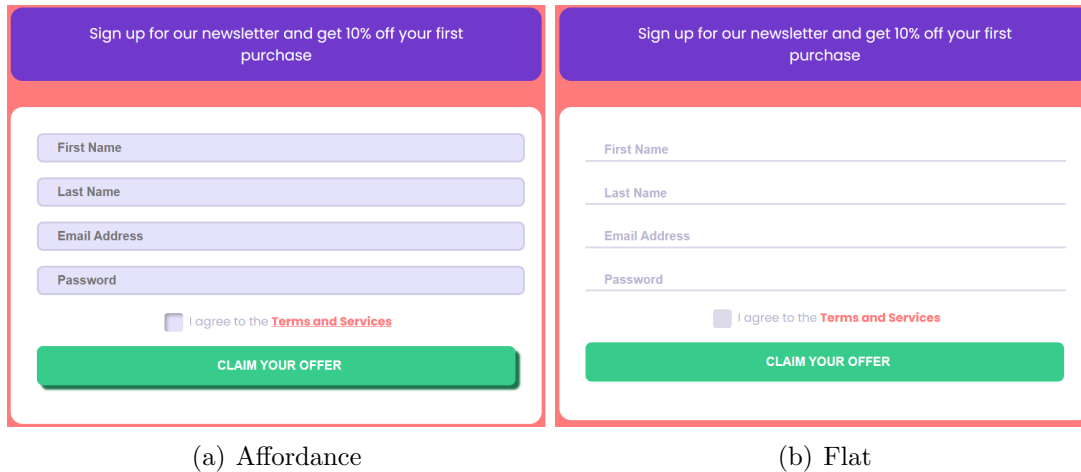


Figure 5.1: Comparison of affordance focused and flat design.

of each website. Using these two versions I compare the time a user takes to find every interactive element on the unmodified website with the time it takes them on the modified version.

The selected templates vary in complexity and purpose from old and simple looking text paragraphs in front of a white background to modern looking websites with an integrated shop. My modifications range from added shadows and borders to underlining and changing the color of links. In some occasions I also modify the original template to better match the flat design principles or to limit the scope of the website. Figure 5.1 shows a section of one of the websites in both the modified high affordance and the unmodified flat design version. More example comparisons between both versions are shown in the appendix in Section A.2.

To simplify the task for the participants of my experiment, I include a feedback box to every website. The feedback box shows the amount of interactive elements on the website and how many the participant has already clicked. Whenever the participant finds a new interactive element and clicks it, it gets marked by changing color. Multiple buttons of the same type get marked at once if one of them is clicked. When all interactive elements are clicked the next website opens automatically. A participant clicks through all five websites. This always includes at least two unmodified and two modified versions. After completing the task a file with the total amount of clicks and the time spent on each website is downloaded.

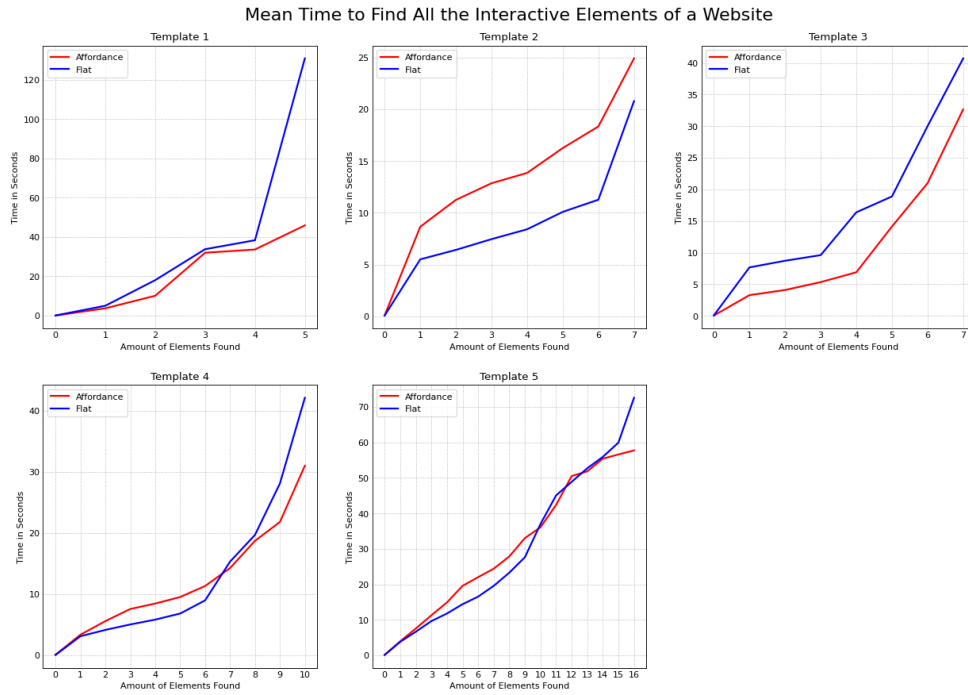


Figure 5.2: Comparison of the time needed to find all elements.

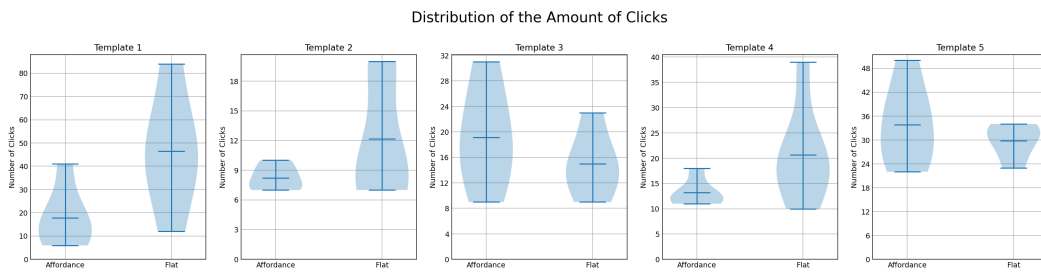


Figure 5.3: Comparison of the amount of clicks to find all elements.

5.3 Results

Figure 5.2 shows the mean time it took the participants to find the interactive elements of each template. The blue lines represent the unmodified flat versions and the red ones the modified high affordance versions. In Figure 5.3 the distribution of the total amount of times the participants clicked before finding all interactive elements is displayed.

A total of ten participants completed this experiment, with each version of a website being completed five times. Comparing the times to find all interactive elements of the different versions with the Mann-Whitney-U-Test results in the following p-values: 0.004, 0.429, 0.792, 0.662 and 0.177 in the numerical order of

the templates. The p-values of the click amounts are: 0.052, 0.350, 0.360, 0.119 and 0.647.

5.4 Discussion and Interpretation

The results of the experiment show no significant difference in completion time and click count between both design choices except for Template 1. In the flat version of Template 1, it took all the participants more than 50 seconds to find the last interactive element. It was always the same link, that was at the very bottom of the page and is visually indifferent to plain text. It is only recognizable as an interactive element when hovered or by realizing it has the format of an URL and thus assuming that it is interactive.

Beside that one link, with an extremely low amount of affordance in the flat design version, the two design types seem to be similar in terms of usability. Nowadays, most users are familiar with the general types of designs and structures most websites use. They do not look for shadows or other small pieces of affordance to determine if an element is interactive or not. From observing how the participants scanned the websites for interactive objects, I would say all they look for is any visual contrast to the plain text on the website. The simplest way to achieve such contrast is by giving interactive elements a consistent color throughout the website.

I need to add that my conclusions, based on the results of this experiment, might be wrong, as there are some flaws in the setup of the experiment. The biggest one being that the participants do not interact with the websites in a natural way. They were given the task to find interactive elements and thus have all their focus on looking for them. This is not usually the case when browsing websites, the focus is on finding some information or completing a different task not on finding all buttons.

Chapter 6

Dialog Feedback

6.1 Underlying Concepts

6.1.1 Placement in Human-Computer Interaction

To create a good user experience we need to bridge the gulf of evaluation introduced in Chapter 2. The main design principle to bridge the gulf of evaluation is Feedback. There are many ways in which a system can give feedback. This also means that there are a lot of mistakes one can make when implementing feedback. In this chapter we talk about general rules for feedback and will make a direct comparison between modal and modeless dialog boxes.

6.1.2 Rules for Feedback

A user should get feedback on every action they take. This is easy to accomplish, the tough part is doing it without unnecessarily interrupting their workflow and annoying them. Every time we throw a dialog at the user we take their attention away from their initial task. This needs to be carefully considered to choose the type of feedback that takes the appropriate amount of the user's attention.

Another act of balance is the amount of pop-ups a system sends out. If pop-ups appear regularly users will start to click them away before reading the feedback. This makes them effectively non-existent and they will only act as another click the user has to make.

The timing of feedback is also highly relevant. After any interaction from the user the system should respond within a second. Whenever the system does not act upon an input within this time-frame the user will start questioning if they did

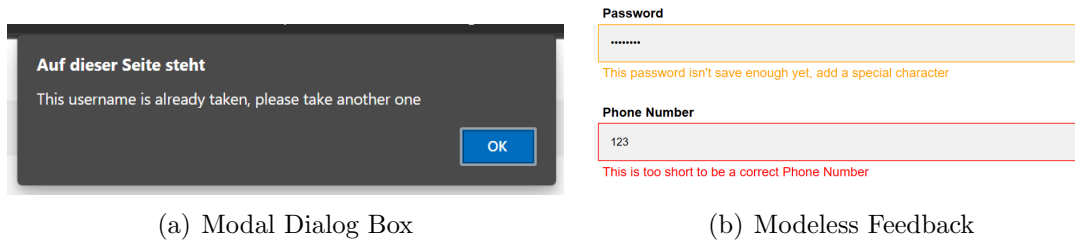


Figure 6.1: Example feedback given from the forms of the experiment.

a mistake. Another part of the timing is which action should trigger the feedback. When filling out a form should we interrupt the user immediately after entering an incorrect input or should we wait until they fill out the whole form and submit it. He might realize the mistake by himself and thus we do not need to interrupt his workflow. The general answer is, make it impossible for the user to give invalid inputs whenever you can. For example do not allow letters when asking for a number.

6.1.3 Two types of Feedback

Modal Feedback If feedback forces you to interact with it, it is modal. This makes it impossible for the user to continue the initial action before closing the dialog box with the feedback (Figure 6.1(a)). This is the most common type of feedback although it is rarely the best option.

Modeless Feedback Alan Cooper's concept of modeless feedback [11] does not require an interaction. The application does not freeze and the user is not forced to interrupt his workflow. Modeless feedback often is visible all the time and updates whenever the user makes a change (Figure 6.1(b)).

6.2 Experiment Setup

In this experiment participants will fill out two forms that give different feedback. This allows me to evaluate the difference of using modal or modeless feedback in forms. Is modeless feedback strictly superior when implemented according to the general rules of feedback?

Both forms consist of the same eight input fields and two buttons. The forms simulate the creation of a new account and ask for the usual data needed in this process. Both versions have the same design they only differ in the feedback they

give on invalid inputs. In addition while filling out the forms a simulated windows message pops up in the respective feedback type.

To force wrong inputs, the participants are given a sheet with the data they are supposed to put into the respective fields. On both data-sets three entries are invalid. Participants need to replace these entries with their replacement option given in the data-set. A replacement option is given to all entries, to ensure that participants do not know which entries are wrong before getting the feedback.

The modeless feedback version has a constraint on the submit button until all fields are filled correctly. Wrong inputs get immediately marked with a red border and have an explanation why the input is invalid directly below them. Once the input is corrected the red border and explanation are removed. The simulated windows message is that the battery is on 20%, it is visible in the bottom right corner and disappears after 10 seconds without any interaction.

The form with modal feedback has no constraint. Feedback on wrong inputs is given once the submit button is clicked. A modal dialog box with the explanation to the first invalid entry will pop up. After closing the modal dialog box, the participant can edit the affected entry and submit again. This process repeats until all entries are corrected. The simulated windows message is a system update that will start automatically in 30 minutes. This time it needs to be clicked away before the participant can proceed to fill the form.

On top of tracking the time participants take to fill out the forms they are also asked to answer a survey. The survey contains three questions, two about their user experience with the implemented feedback and one that checks if they read the simulated windows message. The last question serves the purpose to check whether the forced interruption of the users workflow makes it more likely that they notice and read the message.

6.3 Results

All eight participants completed the modeless form faster than the modal one. The distribution of completion times can be seen in Figure 6.2. The modeless version took an average of 65 seconds to fill correctly while the modal version took 89 seconds on average. Using the Mann-Whitney-U-Test to evaluate the statistical significance of these results we get a p-value of 0.195.

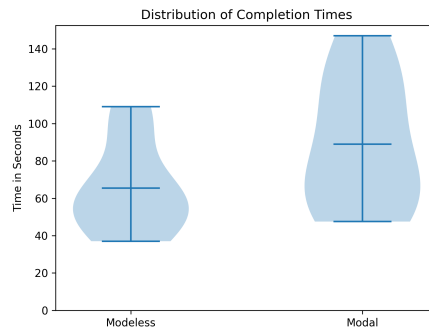


Figure 6.2: Comparison between the completion times of the two forms.

In the survey six out of the eight participants stated they had a bad user experience with the modal feedback version of the form. In the modeless feedback one, all eight participants were happy with the user experience. Neither version lead to a situation where the user felt lost and did not know what to do anymore.

Regarding the simulated windows messages, in the modal version all eight participants noticed the message and were able to recall the important parts of it. In the modeless version the simulated message went completely unnoticed by two of the eight participants. The ones that noticed it were also able to recall the information from the message.

6.4 Discussion and Interpretation

Although the statistical evaluation of the results shows, that the completion times on the modeless version might only be lower because of chance, I do believe that the modeless feedback improves both user experience and efficiency. It allows the users to take full control of the workflow and they can adapt to feedback at a time that suits them. This way they can choose an efficient way to resolve issues.

Feedback that is not related to the current task should not be modeless, as it might just get overlooked. If it being overlooked is acceptable, then it is probably not important enough to justify to take away the users attention. Thus the feedback should be given at another time. If it being overlooked is not acceptable, it should force a user interaction to avoid going unnoticed. In this case, make sure to give the feedback a clear title. Because some users will try to go back to their original task as fast as possible and wont read an entire paragraph before clicking the feedback away.

Chapter 7

Mode Errors

7.1 Underlying Concepts

7.1.1 Placement in Human-Computer Interaction

In this chapter we take a look at human errors and their influence in HCI. Humans will make mistakes and systems need to be prepared to handle them. If one mouse-slip leads to a catastrophic failure, it is not the user that failed but the designer. To create systems that allow for user errors we need to understand and predict possible user errors.

7.1.2 Error Types

Neville Stanton has assembled a lot of previous research on human errors and classifies human errors into three types [12]. The three high-level types of error are slips, lapses and mistakes. Slips and lapses are unintended actions the user makes because of attention and memory failure respectively. Contrary a mistake is an intended action from the user. In a mistake the user failed in the assessment and/or planning of his action not in the execution. The opposite applies for slips and lapses, the user's plan is correct but he fails to execute it correctly.

All three types can be further distinguished and placed into subcategories. One subcategory of a lapse are mode errors. The experiment of this chapter is based on this subcategory and thus the remaining of this chapter will focus on it.

7.1.3 Mode Errors

Mode errors occur when a user has a correct plan and does the right action but is not aware that he is in the wrong mode. This requires the system to have

multiple modes it can be in. Having multiple modes allows for a big amount of possible actions while keeping the single actions rather simple. In a system with multiple modes the same user interaction leads to different outcomes depending on the current mode of the system.

Drawing tools have a lot of different modes. Usually they are represented in a tool bar where the current mode, in this case the tool, is highlighted. A mode error in a drawing tool would be to try to do something that requires one specific tool while another tool is activated. Such an error occurs when the user forgets which mode the system is in, thus it is a failure of the user's memory.

With the knowledge of why and when errors occur, we can design our interfaces to prevent or at least handle them. To prevent mode errors we need to assist the user's memory such that he does not forget which mode the system is in. This is done with the feedback the user gets from the system.

7.2 Experiment Setup

In this experiment the goal is to create an interface to a drawing tool that prevents mode errors. I try to do this by the use of modeless feedback, look at Chapter 6 for the definition of this concept. To evaluate the success of my interface I compare it to a second interface without any additional feedback. After conducting the experiment with both versions we will see whether or not the amount of mode errors reduced.

The current mode of the drawing tool is defined by the combination of the three parameters color, size and tool. The color can be set with one of the 14 color icons. The size is set with a slider and the tool can be set to pen or fill with their respective buttons. Other than that there is the canvas the user can draw on, a clear and an undo button.

In the good interface, the one with modeless feedback, the cursor is replaced with a custom one that adapts to the current mode. The modified cursor is a circle with the size of the current pen and filled with the currently selected color. When the fill tool is active the icon of the fill button is visible next to the cursor its size is fixed, as the size setting does not matter for the fill tool. With this modified cursor the current mode is always shown directly at the center of the user's attention. This



Figure 7.1: A replication done with the good interface.

feedback allows the user to not rely on his memory and thus should reduce the amount of mode errors.

To carry out the experiment all participants replicate the same given image with the drawing tool. The chosen image intentionally requires a lot of switching between modes (Figure 7.1). A participant will replicate the image with both versions of the drawing tool and then rate both versions from one to five. The process of replicating the image is screen recorded. Thus the amount of mode errors can be counted afterwards. Participants are not told what will be evaluated beforehand. This way we can simulate a more natural interaction with the drawing tool.

7.3 Results

With the normal, unmodified cursor the seven participants did an average of 3.3 mode errors while replicating the image. Out of the 23 total mode errors 18 were the participant completely forgetting that they need to change the mode. The other five were caused by a slip that went unnoticed and then led to a mode error afterwards on. The most common mode error was forgetting to turn back into pen mode after using the fill tool, this happened 12 times. As shown in Figure 7.2 every participant made at least one mode error with a maximum of six mode errors. The modified cursor lead to a total of zero mode errors between all seven participants. They still forgot to switch modes at times or slipped in the process of switching them, but they always noticed before restarting to draw.

Comparing the amount of mode errors with the Mann-Whitney-U-Test results in a p-value of 0.001. Meaning that the difference between both versions is statistically significant.

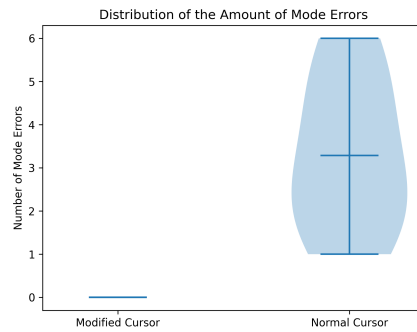


Figure 7.2: Results Mode Error Experiment

In the survey, the good interface got an average rating of 3.4 out of five on how easy it is to replicate the image with the given drawing tool. The bad interface got a slightly worse average score of 3.0 out of five. No participant gave the version without feedback a better rating than the one with additional feedback. The ratings have a p-value of 0.506 on the Mann-Whitney-U-Test and thus the difference is not of statistical significance.

7.4 Discussion and Interpretation

The results show that mode errors can be significantly reduced or even eliminated with good modeless feedback. While the difference in the ratings from the survey is not significant, I do believe that the effort put into the additional feedback significantly improves the interface and thus the whole drawing tool. I think the difference in the ratings is rather small because of how the question in the survey was phrased. It asks for a general rating on the difficulty of the task instead of a more specific question about the feedback or mode errors.

After the evaluation of the screen recordings I conclude that the placement of the modeless feedback is the most important aspect. The users still skipped the process of switching modes at times, but with the current mode directly displayed on the cursor, their focus of attention, they realised they forgot something before causing an error. When designing an interface with multiple modes try to display the current mode, in a modeless manner, at the point where users will be focused at to complete their action. Users will not actively look for feedback before causing mode errors, as they are not aware of their lapse beforehand.

Chapter 8

Conclusions and Future Work

8.1 Overall Conclusion

Here I will make a general conclusion to the overall objective of this thesis. For more precise conclusions and reevaluations of the specific HCI theories take a look at the discussion and interpretation sections of the respective theory.

One objective of this bachelor thesis was to find out how well the existing laws, rules and principles of HCI align with the reality of human-computer interactions. After conducting six experiments on HCI theories I can say that all of them are applicable in practice. The process of applying them sensibly however, is not always as straight forward as one might think it is.

Most theories are simplified and ignore some of the influences. Because of that, it is important to think about the specific circumstance of your application before blindly following all rules. In Chapter 2 we have seen that when choosing button placement and size, there is more to think about than just following Fitts' Law to minimize the movement time. You also need to make sure that users will find the button quickly or the impact of mouse slips.

The good thing about this process is that HCI theories give you a good starting point. By then planning and analyzing the implementation of the theory you will learn more about your application and be able find a suitable solution. Meaning that even if a theory does not align perfectly with the reality of human-computer interactions, it is still helpful in the design process.

Another objective was to find out where these HCI theories should be further improved. As mentioned above the weakness of most theories lies in their simpli-

fication or generalization. Thus to improve them, we could give more context of the specific circumstances in which the theory applies. While this would make the application of the theories easier it would also make the theories much longer and less pretty. This might not sound as important but it is, as being general and short is a key aspect of these HCI theories. In order to balance out both these benefits, I would suggest to add a list to each theory of other influences one should think about when applying this theory.

Further improvement on HCI theories can be made by updating them to the newest technologies and the shifts in user behavior. As seen in Chapter 5, theories might be outdated due to changes in either the computer or human element of the interaction. In this case the theory did not apply as the users already came into the interaction with much more knowledge and familiarity to the digital world than assumed when creating the theory.

The last motivation for this thesis was to create experiments that can be used to show the importance of HCI. I believe that the experiment about Shneiderman's resource model from Chapter 3, the comparison between recognition and recall in the experiment of Chapter 4 and the experiment about mode errors in Chapter 7 all do this very well. The experiment about dialog feedback from Chapter 6 works for showing the impact of good feedback, but I find it to be a bit lackluster and thus not ideal to advertise HCI with. The experiments about Fitts' Law and Digital Affordance in Chapters 2 and 5 can not be used, as their results do not match the theories. The experiment about Fitts' Law might work well after changing it to not include the time it takes to spot the target.

8.2 Future Work

This thesis did not have the objective to come up with new theories. Thus the scope of future work based on this thesis is very limited. One could fix the flaws in my experiments to confirm or reject the conclusions that are not based on data as the flaws did not allow to do so. Similarly one could increase the sample size of experiments that did not contain statistically significant differences in their results.

There are many other theories from the field of HCI that I did not get to test and reevaluate. It might be worth to look through them and check if they are still up do date. As shown with the experiment of Chapter 5 some theories might be outdated. You can then try to adapt the theory to better suit current technologies

and user behavior. This adaptation of theories is not only limited to outdated ones, some simply never included the way we interact with the newest technologies.

Lastly one can always come up with new laws, design principles and recommendations. Think of an influence to human computer interactions that is not widely considered yet. Get knowledge from other research fields and explain its influence to HCI. It will always be beneficial to improve the usability of our technologies.

Appendix A

Appendix Title

A.1 More Plots to the Experiment on Fitts' Law

Figure A.1 shows scatter plots that compare the results of targets with the same size and input device. Thus they show the impact of the targets location. Only targets with a bigger distance than 1000 pixels are included to make sure to only include edge targets on a different edge to the previous target. This is done because having two targets on the same edge one after each other, will take away the possible benefits of the edge for the second target. Some numeric data and the p-value to this comparison is shown in Table A.1.

Figure A.2 shows similar scatter plots. This time comparing the results of big targets on the edge with small targets in the center using the same input device. Again only targets with distances bigger than 1000 pixels are included. The numeric and statistical evaluation is shown in Table A.2

Figure A.3 shows the distribution of the amount of missed clicks. Each plot compares the amount of missed clicks of the same target location with the use of the same input device. Again the averages and the p-value is shown in Table A.3.

Version	avg Time Center	avg Time Edge	difference	p-value
Mouse Big	0.96	1.02	0.06	0.188
Touch-pad Big	1.27	1.36	0.10	0.082
Touch-screen Big	0.72	0.94	0.22	0.000
Mouse Small	1.19	1.41	0.22	0.009
Touch-pad Small	1.40	1.73	0.34	0.000
Touch-screen Small	0.87	1.19	0.31	0.000

Table A.1: Comparison between average time to hit center and edge target.

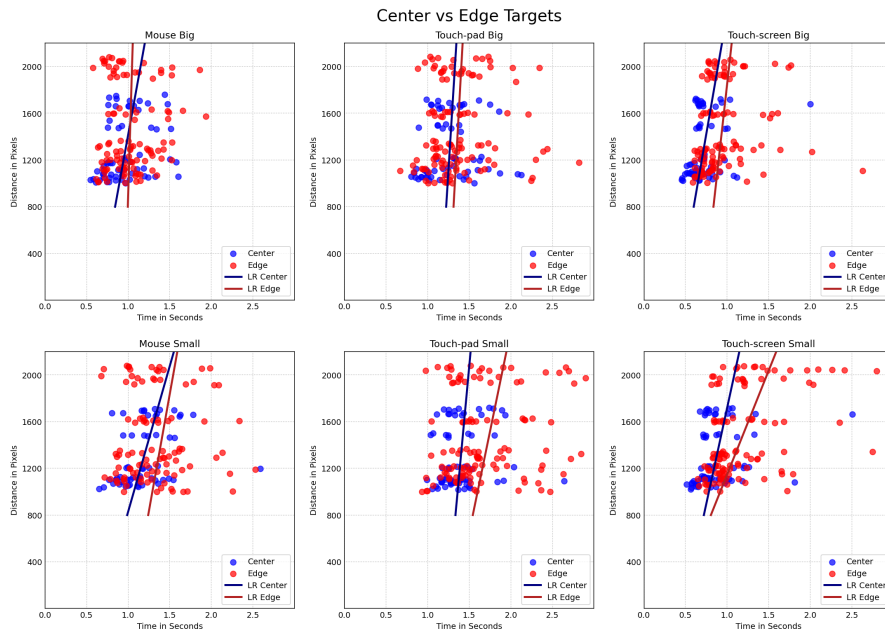


Figure A.1: Comparison between center and edge targets.

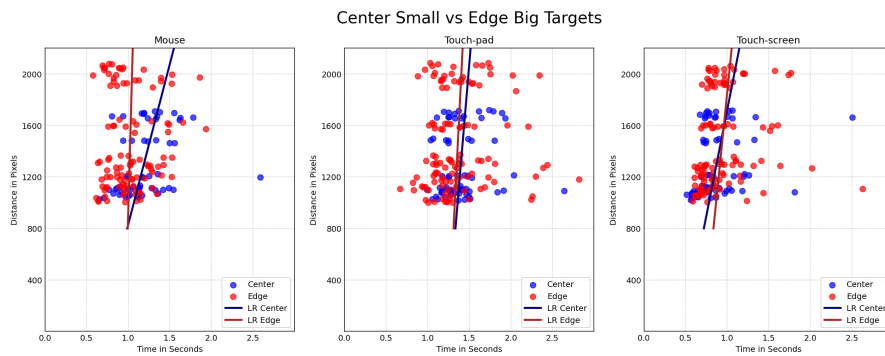


Figure A.2: Comparison between small center and big edge targets.

Version	avg Time Center	avg Time Edge	difference	p-value
Mouse	1.19	1.02	-0.16	0.001
Touch-pad	1.40	1.36	-0.04	0.498
Touch-screen	0.87	0.94	0.07	0.174

Table A.2: Comparison between average time to hit a small center and a big edge target.

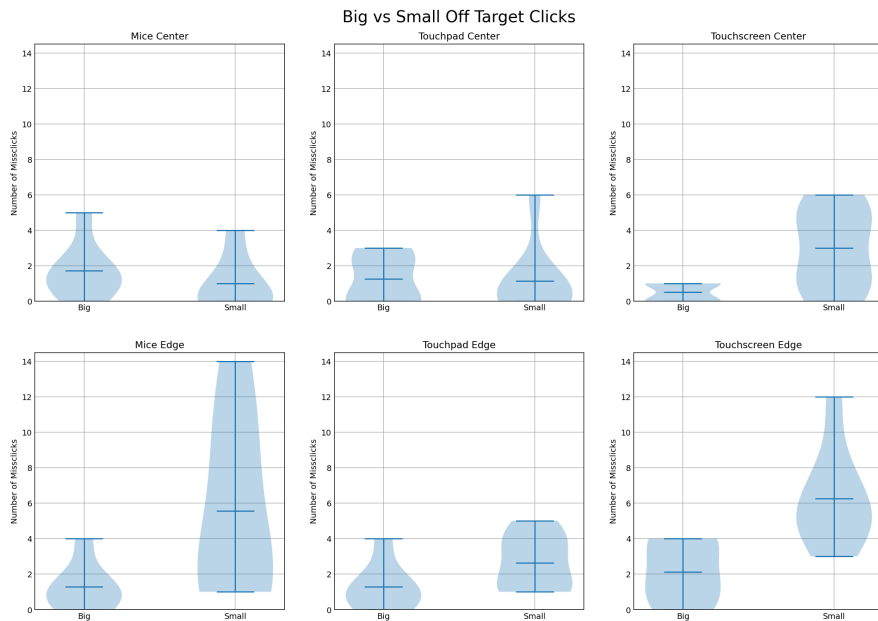


Figure A.3: Comparison of the amount of missed clicks.

Version	avg Errors Big	avg Errors Small	difference	p-value
Mouse Center	1.71	1.00	-0.71	0.259
Touch-pad Center	1.25	1.13	-0.12	0.693
Touch-screen Center	0.50	3.00	2.50	0.018
Mouse Edge	1.29	5.57	4.28	0.058
Touch-pad Edge	1.29	2.63	1.34	0.105
Touch-screen Edge	2.13	6.25	4.12	0.002

Table A.3: Comparison between average missed clicks while clicking 30 small and 30 big targets.

A.2 Websites in Affordance and Flat Design

In the following pages you can find snippets of the websites used in the affordance experiment. A snippet to Template 2 is shown in Chapter 5 itself. Every snippet is shown in both design versions, first in the high affordance design and then in flat design.

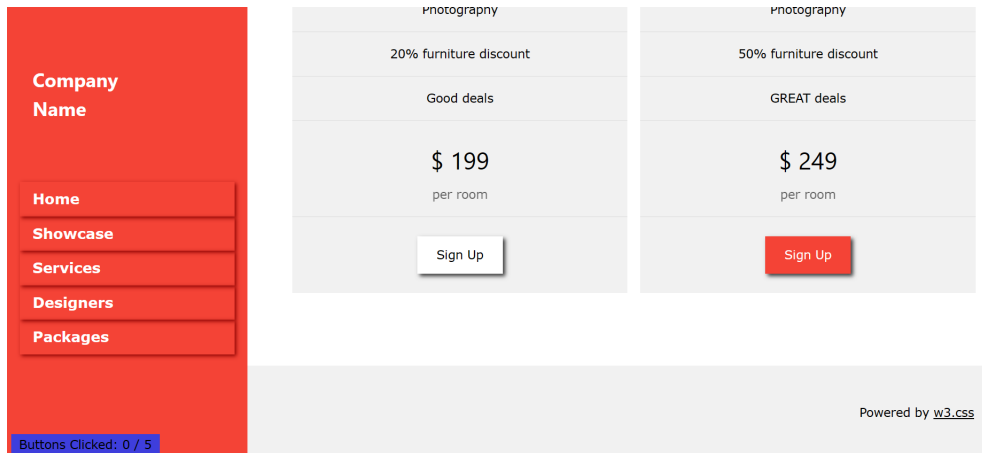


Figure A.4: Snippet from Template 1 in affordance design.

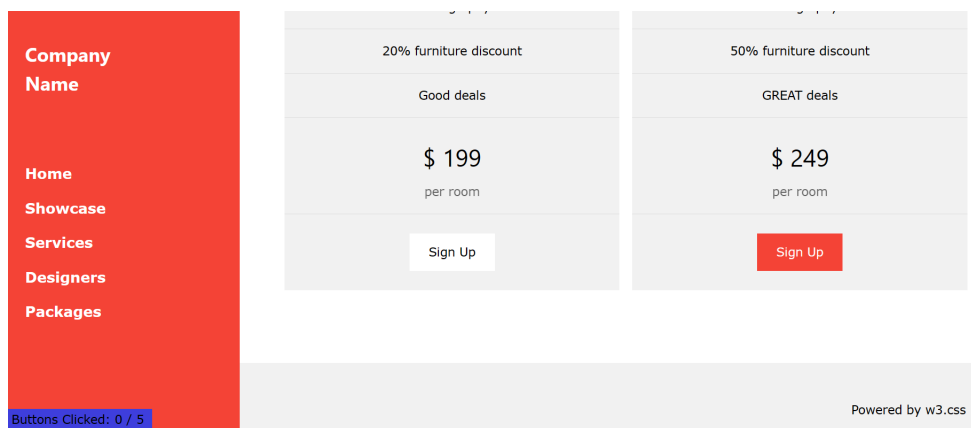


Figure A.5: Snippet from Template 1 in flat design.

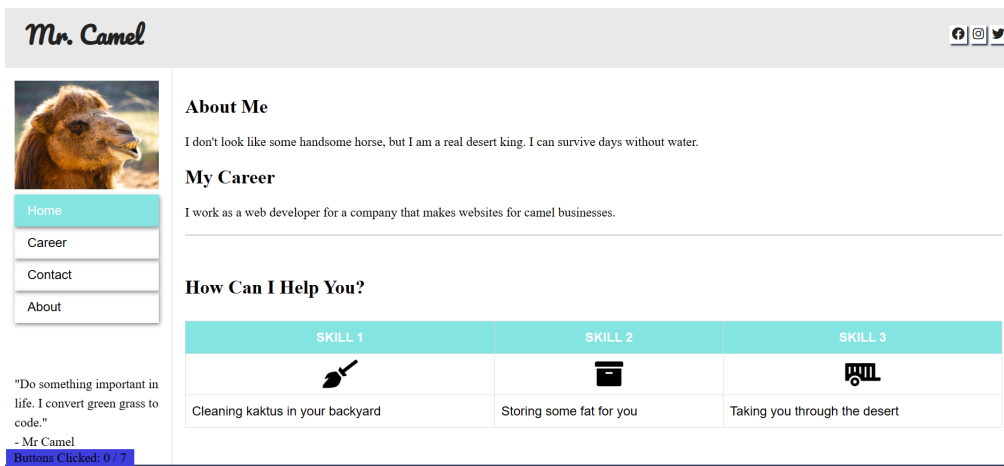


Figure A.6: Snippet from Template 3 in affordance design.

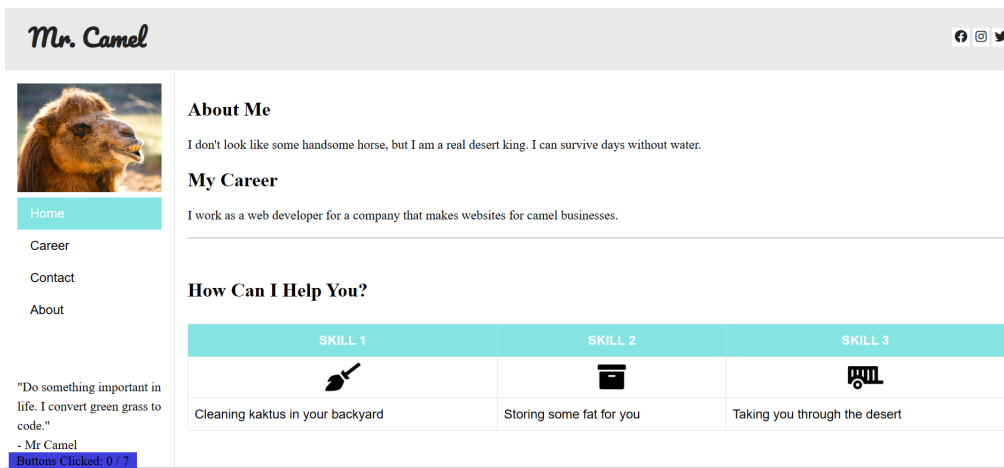


Figure A.7: Snippet from Template 3 in flat design.

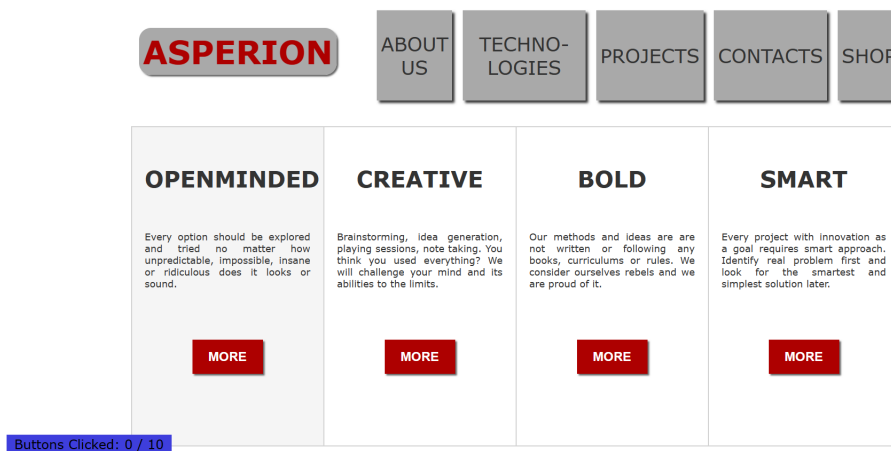


Figure A.8: Snippet from Template 4 in affordance design.

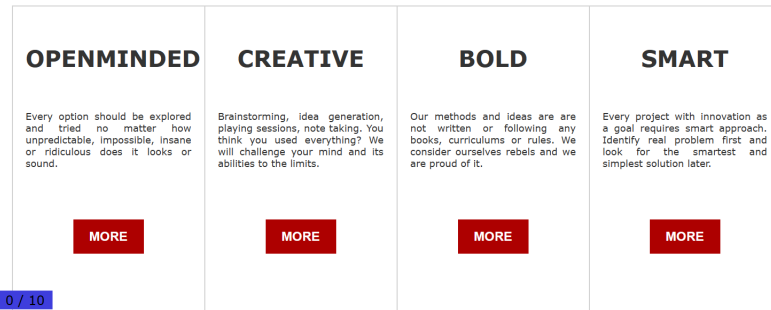


Figure A.9: Snippet from Template 4 in flat design.

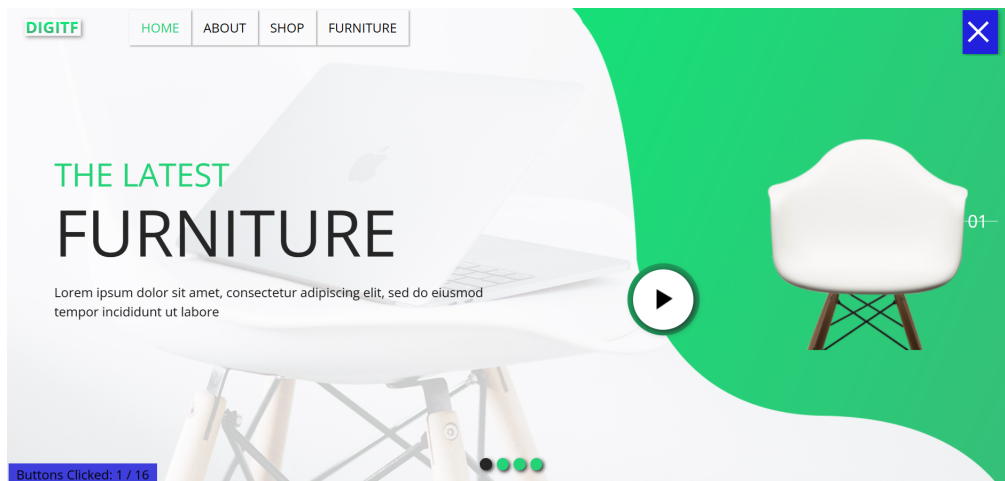


Figure A.10: Snippet from Template 5 in affordance design.

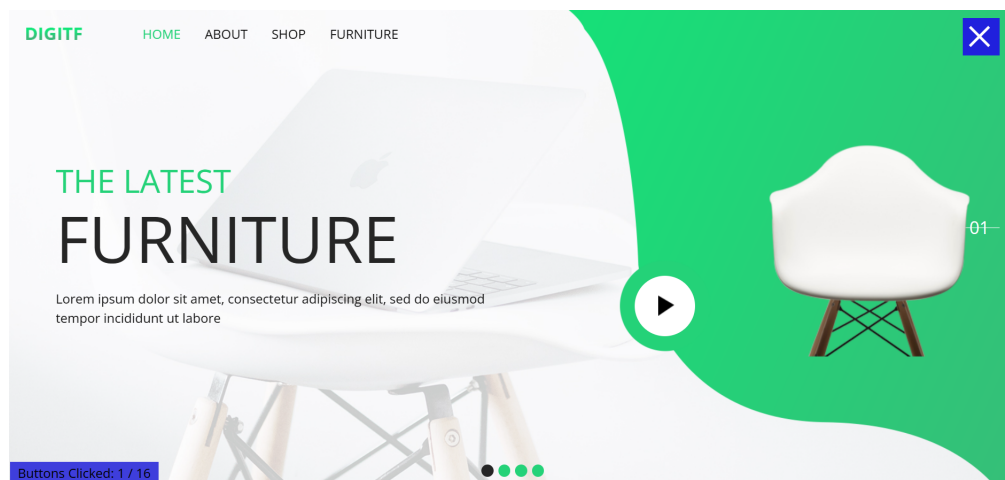


Figure A.11: Snippet from Template 5 in flat design.

Bibliography

- [1] Thomas T Hewett, Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. *ACM SIGCHI curricula for human-computer interaction*. ACM, 1992.
- [2] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Hum. Comput. Interact.*, 7(1):91–139, 1992.
- [3] Digby Elliott, Werner Helsen, and Romeo Chua. A century later: Woodworth's (1899) two-component model of goal-directed aiming. *Psychological bulletin*, 127:342–57, 06 2001.
- [4] Ben Shneiderman. The limits of speech recognition. *Commun. ACM*, 43(9):63–65, 2000.
- [5] Jessica Chen, David Lyell, Liliana Laranjo, and Farah Magrabi. Effect of speech recognition on problem solving and recall in consumer digital health tasks: Controlled laboratory experiment. *J Med Internet Res*, 22(6):e14827, Jun 2020.
- [6] R.C. Atkinson and R.M. Shiffrin. Human memory: A proposed system and its control processes. In Kenneth W. Spence and Janet Taylor Spence, editors, *Psychology of Learning and Motivation*, volume 2 of *Psychology of Learning and Motivation*, pages 89–195. Academic Press, 1968.
- [7] Alan Baddeley. Working memory: Theories, models, and controversies. *Annual Review of Psychology*, 63(1):1–29, 2012. PMID: 21961947.
- [8] Gamze Kayacan, N. Nazlı Ateşgöz, and Uğur Sak. A comparative analysis of psychometric properties of memory tasks and their relationships with higher-order thinking skills: Recognition versus recall. *Talent*, 10(2):162 – 175, 2020.
- [9] Donald A. Norman. *The Design of Everyday Things*. Basic Books, New York, 2 edition, 2013.

- [10] Mo Chen, Georges M. Fadel, and Ivan Mata. Applications of affordance and cognitive ergonomics in virtual design: A digital camera as an illustrative case. *Concurr. Eng. Res. Appl.*, 30(1):5–20, 2022.
- [11] Alan Cooper, Robert Reimann, and Dave Cronin. *About Face 3: the essentials of interaction design*. Wiley, 2007.
- [12] Neville A. Stanton. *Human Error Identification in Human-Computer Interaction*, page 371–383. L. Erlbaum Associates Inc., USA, 2002.

Erklärung

gemäss Art. 30 RSL Phil.-nat.18

Name/Vorname: Gaumez Julien

Matrikelnummer: 20-130-688

Studiengang: Computer Science

Bachelor Master Dissertation

Titel der Arbeit: Evaluating Human-Computer Interaction Theories through Practical Experiments

LeiterIn der Arbeit: PD Dr. Kaspar Riesen

Ich erkläre hiermit, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe r des Gesetzes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist. Für die Zwecke der Begutachtung und der Überprüfung der Einhaltung der Selbständigkeitserklärung bzw. der Reglemente betreffend Plagiate erteile ich der Universität Bern das Recht, die dazu erforderlichen Personendaten zu bearbeiten und Nutzungshandlungen vorzunehmen, insbesondere die schriftliche Arbeit zu vervielfältigen und dauerhaft in einer Datenbank zu speichern sowie diese zur Überprüfung von Arbeiten Dritter zu verwenden oder hierzu zur Verfügung zu stellen.

Albligen, 16.06.2023

Ort/Datum

Unterschrift

J. Gaumez