

Praxisprojekt
Titel

von

Student - 11098765

an der Technology, Arts and Sciences TH Köln
Campus Gummersbach
im Studiengang Medieninformatik (Bachelor)

Betreuer: Prof. Dr. Christian Kohls

Technology, Arts and Sciences TH Köln

Technology, Arts and Sciences TH Köln, 8. Dezember 2016

Abstract

Hier steht später das Abstract.

Inhaltsverzeichnis

Abstract	i
Abbildungsverzeichnis	iii
Quellcodeverzeichnis	iii
Abkürzungsverzeichnis	v
1 Interaktive-Elemente	1
1.0.1 Buttons	3
1.0.1.1 Web	3
1.0.1.2 Android	4
1.0.1.3 iOS	5
1.0.2 Input Fields	5
1.0.2.1 Web	6
1.0.2.2 Android & iOS	6
1.0.3 Select & Radio Buttons	7
1.0.4 Links	7

Abbildungsverzeichnis

Quellcodeverzeichnis

Abkürzungsverzeichnis

AWT	Abstract Window Toolkit
API	Application Programming Interface
EDT	Event Dispatch Thread
F3	Form Follows Function
HLS	HTTP Live Streaming
RIA	Rich Internet Application
UI	User Interface

Kapitel 1

Einleitung

1.1 Einleitung

Die nachfolgende Dokumentation beschäftigt sich mit der Konzeption eines Tools, das dem Nutzer dabei helfen soll, grundlegende Gestaltungsregeln zu beachten. Prägend für die Idee des Projektes ist dabei der Aufbau des Studiengangs "Medieninformatik" an der technischen Hochschule Köln. Dieser besteht zu großen Teilen aus Informatikmodulen, beinhaltet jedoch auch Module, die sich beispielsweise mit den Grundlagen der visuellen Kommunikation beschäftigen. Diese Grundlagen finden später in vielen studentischen Projekten keine Anwendung oder zeigen deutliche Mängel.

Das konzipierte Tool soll die nötigen Inhalte interaktiv vermitteln, den Nutzer also bewusst Fehler machen lassen und ihn auf diese Fehler aufmerksam machen. Weiterhin soll das Tool erläutern, warum die vom Nutzer gewählte Lösung nicht optimal ist.

1.2 Relevanz

Zunächst soll hier jedoch die Relevanz eines solchen Tools diskutiert werden. In vielen der Module, die nicht explizit den Bereich Design zum Thema haben fließt die Gestaltung nicht mit

in die Gesamtnote ein. Es stellt sich also die Frage, warum Studenten Zeit in die Gestaltung eines Artefaktes investieren sollten, wenn dadurch kein expliziter Mehrwert entsteht.

Zur Beantwortung dieser Frage sei zunächst das Konzept des *confirmation bias* (zu deutsch etwa "Voreingenommenheit") erläutert. Raymond S Nickerson beschreibt dieses wie folgt:

"Confirmation bias, as the term is typically used in the psychological literature, connotes the seeking or interpreting of evidence in ways that are partial to existing beliefs, expectations, or a hypothesis in hand." [?]

Menschen verwenden Informationen also so, dass sie als Argumente für ihre bisherige Meinung genutzt werden können und entwerten Argumente, die gegen ihre Meinung sprechen. Demnach lassen sich Menschen, sobald sie sich eine Meinung gebildet haben, nur schwer vom Gegenteil überzeugen. Haben Menschen einen positiven Eindruck von etwas, so ist es sogar wahrscheinlich, dass sie später über Fehler hinwegsehen [?]

Eine weiterer wichtiger Aspekt scheint also die Bildung des ersten Eindrucks über eine Sache zu sein. Lindgaard, Fernandes, Dudek und Brown haben festgestellt, dass Menschen ein Zeitraum von 50ms ausreicht, um sich eine Meinung über eine Webseite zu bilden. [?]

Setzt man diese beiden Elemente in Abhängigkeit wird auch die Relevanz dieses Projektes deutlich: Menschen bilden sich sehr schnell ein Urteil über die Gestaltung eines Artefaktes und dieses lässt sich nur schwer wieder ändern. Die oben dargelegten Quellen lassen den Schluß zu, dass dieser schlechte Eindruck auch auf andere, vielleicht Bewertungsrelevante Kriterien übertragen wird.

1.3 Zielsetzung

Ziel des Tools ist es, in Artefakten eine gewisse gestalterische Grundqualität auf visueller Ebene zu sichern. Dabei sollte es möglichst unabhängig vom Artefakt sein, sich also sowohl auf ein Textdokument als auch eine Android-App anwenden lassen. Mit dieser Interdisziplinarität geht natürlich ein Verlust in Details einher, der bewusst in Kauf genommen wird. Mit der Ambition,

nicht abhängig vom Artefakt zu sein muss das Tools selbst eine generelle Zugänglichkeit bieten und darf sich nicht hinter Beschränkung einer bestimmten Plattform verbergen. Hier liegt die Umsetzung als WebApp nahe. Auch potenzielle Ergebnisse des Tools müssen dabei problemlos auf verschiedene Anwendungsgebiete übertragbar sein. Die Zielgruppe sind hierbei explizit Studenten des Studienganges Medieninformatik an der Technischen Hochschule Köln, generell aber Nutzer, die aus verschiedenen Gründen kein oder nur wenig Wissen und Anwendungserfahrung mit visueller Gestaltung haben.

1.4 Forschungsfrage

Die Forschungsfrage spaltet sich in zwei Teile: Zum Einen müssen relevante Themengebiete mitsamt ihrer Regeln für ein solches Tool erarbeitet werden, zum anderen muss eine Form gefunden werden, in der die erarbeiteten Themen und Regeln dem Nutzer nahegelegt werden sollen. Die ausformulierte Forschungsfrage lautet somit:

“Ermittlung relevanter Themengebiete und Konzeption derer interaktiven Darstellung für die Entwicklung eines Tools zum erstellen von Gestaltungslösungen im Hochschulkontext”

1.5 Vorgehensweise

Als erster Schritt mussten die relevanten Themengebiete definiert werden. Hierfür musste eine Methode zur Ermittlung gefunden werden. Eine genauere Dokumentation dieses Schrittes findet sich in Kapitel 2.

Darauf folgend mussten in den definierten Themengebieten Regeln gefunden werden, die nach Möglichkeit auch programmatisch abgebildet werden können. In diesem Schritt fand sich auch die größte Herausforderung des Projektes. Beim finden von Gestaltungslösungen gibt es oft keinen richtigen oder falschen weg und viele Entscheidungen werden auf einer subjektiven Ebene getroffen. Daher soll das Tool dem Nutzer eher helfen, Grundlagen zu schaffen, als konkrete Lösungen zu finden.

Für jeden dieser Bereiche wurden die Ergebnisse in Mockups oder funktionalen Prototypen festgehalten.

Kapitel 2

Einleitung

2.1 Einleitung

Für die Ermittlung der relevanten Themengebiete musste zunächst eine geeignete Methode gefunden werden, hierfür kamen verschiedene Ansätze in Frage.

Der simpelste Ansatz orientiert sich an den bereits definierten Inhalten eines Buches oder Studienganges, die von qualifizierten Leuten erstellt wurden, zu orientieren. Dieser Ansatz gewährleistet durch die Fähigkeiten der Ersteller zum Einen und die große Auswahl und damit hohe Vergleichbarkeit zum anderen eine gewisse Fehlerfreiheit. Die Inhalte richten sich jedoch an Personen, deren Haupttätigkeit die visuelle Gestaltung ist. Bei Verwendung dieser Methode müssen also die Menge und Tiefe der Inhalte kritisch geprüft werden.

Ein weiterer Ansatz orientiert sich am Workflow der Benutzer und beschäftigt sich eher mit der Frage nach dem "wann" als nach der mit dem "was". Der Ansatz würde sich also daran orientieren, wann ein User eine bestimmte Tätigkeit ausführt und welche Bereiche der visuellen Gestaltung an dieser Stelle einzubringen sind. Dieser Ansatz wurde schnell als ungeeignet verworfen, da mit ihm ein hoher Aufwand in der Recherche von Arbeitsabläufen der Nutzer einher geht.

Ein weiterer Ansatz war ein Fehlerorientierter Ansatz, für den ich mich schließlich auch entschied. Mit diesem Ansatz würden die Fehler untersucht, die Nutzer häufig machen und die

Inhalte des Tools daraufhin angepasst. Die Wahl dieser Methode liegt gerade deshalb nahe, weil das Tool zunächst im Hochschulkontext Verwendung finden sollte. Durch das Mediawiki des Studienganges Medieninformatik war der Zugriff auf viele Materialien gewährleistet, die auf Fehler untersucht werden konnten.

2.2 Vorgehen

Als erster Schritt wurden Materialien gesammelt. Der erste Anlaufpunkt hierfür war das Mediawiki des Studienganges Medieninformatik. Um zu gewährleisten, dass jedes Modul im Katalog vertreten war, wurden alle Dateien untersucht, die im letzten Jahr hochgeladen wurden. Da nur ein Teil der Module das Mediawiki verwendet, wurde weiterhin auch außerhalb dessen nach Material gesucht, wovon sich das meiste auf GitHub wiederfand. Insgesamt wurde so ein Katalog von 448 Dateien erstellt.

Diese wurden zur Orientierung zunächst grob in vier Kategorien unterteilt:

- Präsentation
- Plakat
- Interaktiv
- Text

Zur Gruppe *Interaktiv* sei hier angemerkt, dass diese bewusst weit gefasst ist. Sie umfasst sowohl Android-Apps, als auch Websites mit Fokus auf Front- oder Backend. Eine weitere Unterteilung wurde an nicht als sinnvoll erachtet, da so zwar Plattformspezifische Fehler besser hätten gefunden werden können, das Tool sich jedoch auf die Grundlagen konzentrieren soll. (Weitere Erläuterungen zu den mobilen Plattformen finden sich in Kapitel 2.3)

Die genauen Zahlen der jeweiligen Kategorien können *Tabelle 1* entnommen werden. Bei der Betrachtung der Werte fällt auf, dass nahezu 50% der untersuchten Artefakte in die Kategorie "Text" fallen. Weiterhin fällt auf, dass Plakate mit etwa 3% aller Artefakte kaum vorhanden waren.

Typ	Anzahl der Dateien
Präsentation	96
Plakat	16
Interaktiv	113
Text	223

TABELLE 2.1: My caption

Das gesammelte Material wurde im nächsten Schritt auf Fehler in der Gestaltung untersucht. Nachfolgend findet sich eine Liste der gefundenen Fehler (in absteigender Reihenfolge nach Häufigkeit ihres Auftretens):

Fehler beim Einsatz von Farben: Es wurden Farbkombinationen verwendet, die flimmerten, visuell anstrengende Farben wurden als Hintergrundfarbe für Texte gewählt, Unharmonische Farbkombinationen, übermäßiger Einsatz von Farben. **Fehlendes Raster:** Beim Ausrichten der Elemente wurde kein Raster verwendet, Elemente wirken dadurch wie zufällig platziert. **Zu wenig Weissraum:** In der Gestaltung wurde zu wenig Whitespace verwendet, die gesamte Gestaltung ist dadurch unübersichtlich und wirkt unruhig. **Visuell anstrengende Tabellen:** Tabellen verwendeten mehr Trennlinien als nötig, verwendeten zu viele/unpassende Farben, es war schwer die Informationen aus der Tabelle zu entnehmen, es wurden Tabellen verwendet, wo keine nötig gewesen wären. **Hierarchie im Text nicht deutlich:** Die Unterscheide zwischen verschiedenen Elementen (z.B. Überschriften verschiedener Ordnungen) waren zu gering, es wurde keine klare Hierarchie deutlich. **Fehlerhafte Ausrichtung von Elementen:** Elemente wurden in sich unsauber ausgerichtet. **Interaktive Elemente nicht deutlich:** Es wurde nicht deutlich, mit welchen Elementen der Benutzer interagieren kann und mit welchen nicht. **Bilder gestreckt/gestaucht/verpixelt** **Inkonsistente Größen:** Gleiche Elemente waren verschieden groß. **Elementgrößen unverhältnismäßig:** Einige Elemente sind im Vergleich zu anderen Elementen unverhältnismäßig zu klein/zugroß. **Zu wenig Kontrast im Text:** Text war wegen des Kontrastes schwer lesbar. **Boxes in Boxes:** Unnötiges Verwenden von Boxen. **Zeilenhöhe zu groß/klein** **Fehler im Textsatz:** Vor allem: Lücken im Blocksatz. **Rechtschreibfehler** **Schlecht lesbare Schriftfamilie**

Abschließend wurde jedem der Fehler eine Oberdisziplin zugeordnet, die somit die Themengebiete definieren, die im Projekt behandelt werden sollen:

* Typographie * Layout * Struktur * Farben * Interaktive Elemente * Tabellen * Bilder

2.3 Kritische Reflexion

Auch wenn dieses Fehlerorientierte vorgehen für das Projekt sehr hilfreich ist bringt es einige Gefahren mit sich.

So ist das Material von nur zwei Semestern keineswegs dazu geeignet, empirische Ergebnisse zu liefern. Die erarbeiteten Ergebnisse liefern lediglich einen Überblick über die Fehler, die in naher Vergangenheit vorherrschend waren. Außerdem ändern sich Inhalt von Modulen häufig von Semester zu Semester, eine Untersuchung im nächsten Jahr würde also mit hoher Wahrscheinlichkeit neue Fehler zeigen.

Außerdem sollten die auf Basis von Fehlern gefundenen Schwerpunkte nicht als exklusiv und vollständig angesehen werden. Es muss darauf geachtet werden, dass ggf. auch andere relevante Themen Einzug erhalten, die nicht explizit Teil der gefundenen Fehlern waren.

2.4 Mobile Plattformen

Die beiden größten Mobilen Plattformen, *Android* und *iOS* besitzen eigene Design Guidelines, die bestimmte Themengebiete von vorn herein stark einschränken. Als Beispiel sei hier die Wahl der Schrift genannt. Sowohl die Guidelines für Android als auch für iOS legen die Verwendung einer bestimmten Schrift nahe. Da sich dieses Tool nicht über die Plattformspezifischen Guidelines hinwegsetzen sollte, muss von Anfang an die Plattform auf der das Projekt des Nutzers laufen soll bekannt sein, die Inhalte des Tools müssen sich entsprechend anpassen.

Kapitel 3

Einleitung

Das Kapitel Typographie könnte als eines der wichtigsten Kapitel dieses Projektes beschrieben werden. Typographie kommt in fast jeder Art von Artefakt vor und bildet den Grundstein einer guten Gestaltung. Gerade weil Typographie aber in so vielen Gebieten Anwendung findet, kann hier unmöglich jeder dieser Anwendungsfälle abgedeckt werden. Das Projekt befasst sich deshalb auf einem grundlegenden Level mit der Typographie und bietet einen Leitfaden zur Erstellung von Fließtexten. Ziel soll es sein, am mit Hilfe des Tools einen gut gesetzten und lesbaren Text zu erstellen. Sehr Graphische Anwendungsfälle wie zum Beispiel Plakate werden hier bewusst nicht angesprochen.

3.1 Schriftarten

Um einen Text zu setzen müssen eine Reihe von Entscheidungen getroffen werden. Die erste dieser Entscheidungen stellt die Entscheidung über die Schriftart dar. Schriftarten werden in Kategorien unterteilt, die nicht immer einheitlich sind. So verwendet Strizver [?] andere Gruppierungen als das "British Standards Classification of Typefaces" [?, S. 51]. Die zwei Gruppierungen "serif" und "serifenlos" finden sich jedoch in jeder Art der Gruppierung wieder und diese sollen auch in diesem Teil des Projektes betrachtet werden. Auch andere Schriftarten haben ihre Daseinsberechtigung und Anwendungsfälle, jedoch werden diese unter der Zielsetzung, einen gut lesbaren Fließtext zu erzeugen hier nicht behandelt.

3.1.1 Serif oder Serifenlos?

Je nach Medium lassen sich in der Lesegeschwindigkeit bei serifen und serifenlosen Schriftarten Unterschiede feststellen, so kommen sowohl Josephson [Josephson, S. (2008). Keeping your readers' eyes on the screen: Ane

Weiterhin hängt die Lesegeschwindigkeit außerdem davon ab, ob der Leser die Schriftfamilie bereits kennt und ob diese explizit für das verwendete Medium entwickelt wurde. [Josephson, S. (2008). Keeping your read

3.2 Schriftfamilien

Wie oben bereits erwähnt gibt es für die Wahl der Schriftfamilie einige Richtlinien, jedoch gibt es keine per se “guten” oder “schlechten” Schriftfamilien. Wenn möglich empfiehlt es sich, einer für das Zielmedium gestaltete Schriftfamilie zu verwenden, wenn die Schnittstelle des Zielmediums ein Bildschirm irgendeiner Art ist.

Es ist weiterhin empfehlenswert, eine Schriftart zu verwenden, die dem Nutzer bereits bekannt ist. Die Webseite [CSS Font Stack](<http://www.cssfontstack.com/>) führt eine Liste mit der Verfügbarkeit von Schriftfamilien auf verschiedenen Betriebssystemen, die einen möglichen Orientierungspunkt darstellt. Nachfolgend ein Auszug aus dieser Liste:

Sans-Serif *Arial(Win:99.84*Tahoma(Win:99.95*TrebuchetMS(99.67*Verdana(Win:99.84*FiraSans([GoogleFonts](<https://www.google.com/fonts/specimen>

Serif *Georgia(Win:99.4*Palatino(Win:99.29*TimesNewRoman(Win:99.67

Monospace *CourierNew(Win:99.73*SourceCodePro([SILOpenFontLicense](<https://github.com/adobe-fonts/source-code-pro>)

3.2.1 Android

Die Material Design Guidelines definieren die Schriftfamilien “Roboto” und “Noto” als Schriftfamilien für die Plattform. Diese sollten für den Nutzer, der ein Projekt auf dem Betriebssystem Android entwickelt, hervorgehoben werden.

3.2.2 iOS

Apple spricht sich für die Verwendung der Systemfont “San Francisco” aus. Auch diese sollte einem Nutzer, der für diese Plattform entwickelt hervorgehoben werden.

3.2.3 Mischen von Schriftfamilien

Beim Mischen von verschiedenen Schriftfamilien entstehen häufig unschöne Textsätze. Beim mischen von Schriftfamilien muss unter anderem darauf geachtet werden, dass die beiden Schriftfamilie miteinander harmonisieren, und sie sich deutlich genug unterscheiden. Die Beurteilung, ob zwei Schriftfamilien gemischt werden sollten findet auf einer sehr subjektiven Ebene statt, daher wird diese im Tool keine Anwendung finden und sich an den Ratschlag von Ellen Lupton halten: > Combining typefaces is like making a salad. Start with a small number of elements representing different colors, tastes, and textures. [Lupton, E. (2014). *Thinking with type*. Chronicle Books.]

3.3 Schriftgrößen

Wichtig für die Lesbarkeit eines Textes ist neben der Wahl der Schriftart- und Familie vor allem die Größe des Textes. Auch hier gibt es nicht die eine, richtige Größe, die Wahl der Größe hängt wie alle anderen Bereich stark von dem Medium ab, in dem die Texte gelesen werden. Einige Eingrenzungen lassen sich dennoch finden, so empfiehlt Runk [Claudia Runk. *Grundkurs Typografie und Layout*: Fr Ausbildung]

Zwar muss die Schriftgröße immer noch individuell angepasst werden, jedoch lassen sich mit diesen Werten bestimmte Grenzen finden. So sind 36px für einen Fließtext im Web wahrscheinlich zu groß und 4pt für einen gedruckten Text zu klein.

3.3.1 Überschriften

Semantisch betrachtet leiten Überschriften einen neuen Sinnabschnitt in einem Text ein. Damit eine Überschrift ihren Zweck erfüllt muss sie einige Eigenschaften besitzen: Es muss deutlich werden, dass mit der Überschrift ein neuer Abschnitt beginnt, sie muss also aus dem normalen

Textfluss heraus fallen. Weiterhin muss deutlich werden, zu welchem Abschnitt die Überschrift gehört.

Um die Überschrift aus dem normalen Textfluss hervor zu heben steht sie klassisch in einer eigenen Zeile. Weiterhin sind sie in der Regel größer als der Fließtext oder (meist bei Überschriften niedrigerer Ordnung) in einem anderen Schriftschnitt gesetzt.

Voraussetzung zum finden einer passenden Überschriftengröße ist also zunächst, dass sie größer sein muss als der Fließtext. Um die Größe der Überschriften zu errechnen bieten sich verschiedene Methoden an.

3.3.1.1 Der goldene Schnitt

Der goldene Schnitt findet in viele Bereichen der visuellen Gestaltung und auch der Natur Anwendung. Er beschreibt ein Verhältnis, das von Menschen in der Regel als harmonisch wahrgenommen wird. Laut [Livio, M. (2003). The golden ratio (p.4). Aryeh Nir Publishers Limited.] wird dieses Verhältnis durch eine unendliche, sich niemals

Als erster Ansatz bietet es sich an, die Größe einer Überschrift zu errechnen, indem die Schriftgröße des Fließtextes mit dem goldenen Schnitt multipliziert wird. Für einen Fließtext mit einer Schriftgröße von 14px würde sich eine Überschriftengröße von $14\text{px} * 1.62 = 22.68\text{px}$ ergeben. Als erster Ansatz scheint diese Methode valide, jedoch bleiben einige Probleme bestehen.

Zum Einen ist 22.68px als Schriftgröße nicht sehr schön. Die Zahl ließe sich aufrunden, aber auch 23px sind als Schriftgröße eher unüblich. Zum Anderen ist mit dieser Methode zwar die Größe für eine Überschrift gefunden, häufig bestehen Texte jedoch aus mehreren Überschriften verschiedener Ordnung. Hier müssen also noch weitere Werte gefunden werden.

3.3.1.2 Typographic Scale

Die Typographic Scale rührt aus den frühen Zeiten des Buchdruckes, als Texte noch aus einzelnen Buchstaben zusammen gesetzt wurden. Die Auswahl an Schriftgrößen war zu dieser Zeit aus rein technischen Gründen limitiert. Eine Variation dieser Typographic Scale kann Abb.X entnommen werden.

[HIERBILDVONSCALEAUSBUCH]

Die Typographic scale könnte also im ersten Schritt genutzt werden, um die mit dem goldenen Schnitt errechnete Zahl zu runden. Die im obigen Beispiel errechnete Zahl lautet 22.68px. Auf der Typographic Scale liegt sie zwischen den Werten 21 und 24. Da er näher an der 24 liegt wird 24px als Schriftgröße für die Überschrift gewählt. Das Ergebnis im gesetzten Text zeigt Abb.X

[HIERERSTESBILDVONGESETZTEMTEXT]

Das Problem der Überschriften verschiedener Ordnungen ist damit jedoch immer noch nicht gelöst. Jedoch kann zumindest eine begrenzte Menge von Zwischenüberschriften aus den Zwischenschritten von Fließtext und der oben errechneten Überschrift gebildet werden. Im Fall des obigen Beispiels wären diese Überschriften 21px, 18px und 16px groß. Ein entsprechend gesetzter Text findet sich in Abb.X.

[HIERZWEITESBILDVONGESETZTEMTEXT]

Befinden sich eine Überschrift und der Text nur 1px oder 2px auseinander, so scheint es ratsam zu sein, die Überschrift auch anderweitig abzuhaben, etwa durch einen fetten Schriftschnitt.

Ein letztes Problem wird deutlich, wenn man die aufgestellte Gleichung für nahe aneinander liegende Schriftgrößen verwendet. So wäre die Überschrift erste Ordnung für Fließtexte der Größe 14px, 16px und 18px jeweils 21px (s. Rechnung)

“ $14\text{px} \cdot 1,62 = 22,68\text{px} \Rightarrow 24\text{px}$ $16\text{px} \cdot 1,62 = 25,92\text{px} \Rightarrow 24\text{px}$ $18\text{px} \cdot 1,62 = 29,16\text{px} \Rightarrow 24\text{px}$ $21\text{px} \cdot 1,62 = 34,02\text{px} \Rightarrow 36\text{px}$ “

Um die Überschriften weiter zu differenzieren wird eine Konstante mit in die Rechnung eingebracht, die zum Wert des goldenen Schnitts addiert wird. Diese errechnet sich aus der Abweichung der Größe des Fließtextes von 14px geteilt durch 10. Dieser Wert hat keine weiteren wissenschaftlichen Belege, vielmehr liefert er für Schriftgrößen zwischen 14px und 24px gut gestaffelte Werte und wird daher verwendet.

Eine Berechnung unter Berücksichtigung der Konstante kann Abb.X entnommen werden.

“ $14\text{px} \cdot 1,62 = 22,68\text{px} \Rightarrow 24\text{px}$ $16\text{px} \cdot (1,62 + 0,2) = 29,12\text{px} \Rightarrow 24\text{px}$ $18\text{px} \cdot (1,62 + 0,4) = 36,36\text{px} \Rightarrow 36\text{px}$ $21\text{px} \cdot (1,62 + 0,7) = 48,72\text{px} \Rightarrow 48\text{px}$ “

Somit wurde ein verwendbarer Ansatz zur Berechnung von Größen von Überschriften basierend auf dem Fließtext gefunden. Dieser Ansatz liefert nur Richtwerte und muss ggf. angepasst werden, weiterhin funktioniert er nur für einen bestimmten Bereich an Schriftgrößen. Für die Verwendung als im Rahmen dieses Tools ist er jedoch ausreichend.

3.4 Abstände im Text

TBD

3.4.1 Zeilenhöhe und -länge

Für Zeilenhöhe und -länge lassen sich die genauesten Richtwerte finden. Im Print-Bereich wird ein Zeilenabstand von 120 [Run11, S. 150] als optimal angesehen, das W3C empfiehlt, mit Blick auf Menschen mit Sehbehinderungen, einen Zeilenabstand von 150 bis 200 [W3C]. Für die Zeilenlänge legen Studien 65 - 75 CPL (Characters per Line) nahe [BFH02], die einen guten Lesefluss ermöglichen, ohne die Suche nach dem Anfang der nächsten Zeile zu kompliziert zu machen. Die Toleranz für die Zeilenhöhe im Tool sollte also bei einem Wert zwischen 120 und 160 liegen. Wie hoch genau eine optimale Zeilenhöhe ist, hängt dabei auch von der Schriftart und dem Einsatzgebiet ab, auch hier kann also nur eine Empfehlung abgegeben werden.

3.4.2 Absätze und Überschriften

Ein Absatz beschreibt einen neuen Sinnabschnitt im Text, der auch visuell erkennbar sein muss. In der Regel werden zwei Absätze dabei durch eine Leerzeile voneinander getrennt. In Textsatzprogrammen reicht diese Richtlinie schon aus und ist selbstverständlich, beim Verwenden von Grafikprogrammen oder beim erstellen von Websites kann aber eine manuelle Einstellung nötig sein. Die Höhe einer Leerzeile lässt sich durch die Zeilenhöhe berechnen:

“ Body: 14px Line Height: 14px * 140 Leerzeile: 14px + 20px = 34px “

Auch die Abstände der Überschriften lassen sich auf Grundlage der Zeilenhöhe berechnen. Als Grundsatz gilt dabei, dass eine Überschrift näher an dem Absatz platziert sein sollte, zu dem sie gehört und dass Überschriften höherer Ordnung mehr Platz haben.

Behalten wir die Größe und Zeilenhöhe des vorherigen Beispiels bei, erhalten wir als Vielfache der Zeilenhöhe 40px, 60px und 80px Spielraum. Diese können nun als obere und untere Abstände der Überschriften verwendet werden:

“ H1: 24px, Top 50px, Bottom 30px (LH * 4) H2: 21px, Top 40px, Bottom 20px (LH * 3) H3: 18px Bold, Top 30px, Bottom 10px (LH * 2) “

Das nachfolgende Bild zeigt einen mit den errechneten Abständen gesetzten Text:

[*HIERNOCHBILD*]

Hier ist erwähnenswert, dass der Großteil aller Programme diese Abstände auch automatisch setzen, wenn die Entsprechenden Elemente auch korrekt ausgezeichnet werden. Wann immer möglich sollte dem Nutzer also zu dieser Methode geraten werden, um die Anzahl von möglichen Fehlerquellen so gering wie möglich zu halten.

3.5 Kontrast

Zuletzt sollte im Tool auch der Kontrast eines Textes behandelt werden, der einen großen Einfluss auf die Lesbarkeit eines Textes hat. Nach Möglichkeit sollte es vermieden werden, lange Textpassagen auch einem Farbigen Hintergrund zu platzieren. Der sicherst Weg wäre ist hier schwarzen Text auf einem weissen Hintergrund. Auch die umgekehrte Zusammensetzung (also Schwarzer Hintergrund und Weißer Text) ist relativ unproblematisch.

Bei allen weiteren Kombination muss der Kontrast überprüft werden. Glücklicherweise ist dies einer der wenigen Bereiche, in denen konkrete Berechnungen eine Kombination als gut oder schlecht einstufen können.

3.5.1 Berechnung

Zur Berechnung werden die in Abschnitt 1.4.3 der Web Content Accessibility Guidelines des w3c definierten Berechnungen verwendet. Diese unterteilen die Kontraste von zwei Farben in drei Level, A (3:1), AA(4.5:1) und AAA(7:1). Das Tool übernimmt diese Einteilung, eine Warnung wird ausgegeben, sobald der Wert für den Kontrast das definierte Level AA unterschreitet.

Anbei findet sich die Umsetzung der in (<https://www.w3.org/TR/2016/NOTE-WCAG20-TECHS-20161007/G18>) definierten Rechnungen in JavaScript, wie sie auch im PoC vorhanden ist.

Kapitel 4

Einleitung

Während der Fehleranalyse konnten immer wieder grundlegende Fehler in der Anordnung von Elementen beobachtet werden, die dazu führten, dass ein Artefakt unsauber wirkte. Diese Fehler lassen sich mit sehr simplen Mitteln wie Hilfslinien oder Grid Systems vermeiden. Hier unterscheiden sich die verschiedenen Medien sowie Plattformen, weshalb dieses Kapitel in jeweils spezifische Unterkapitel unterteilt ist.

4.1 Herleitung

Grid Systems sind sowohl für den Designer als auch für die späteren Nutzer des Artefaktes von Vorteil. Sie helfen dabei, Elemente innerhalb einer Gestaltung sauber anzuordnen und klare Linien zu erzeugen. [Hier noch verweis darauf, dass das menschliche Gehirn gerne nach Mustern sucht]

4.2 Grid Systems im Web

Für Gestaltungen im Web sind Grid Systems sehr verbreitet, nicht zuletzt wegen ihrer hohen Flexibilität im Bezug auf Responsive Webdesign. Wegen der hohen Verbreitung liegt es nahe, dass das Tool kein eigenes Grid System erstellt, sondern nur die Grundlagen vermittelt und auf

eine bereits bestehende Lösung verweist. Hierfür wurden zunächst verschiedene Grid Systems gesichtet und evaluiert.

[gridpak](http://gridpak.com/) Geeignet, um individuelle Grid Systems zu erstellen. Sehr simpel, da nur drei Werte verändert werden müssen (plus Option zum Anlegen von Breakpoints). **Output:** Verschiedene CSS und JS Files und das Grid System als png.

[960px Grid](http://960.gs/) Festes, 960px breites Grid System, keine individuellen Einstellungen mehr Möglich. **Output:** Verschiedene CSS Dateien sowie Vorlagen für viele Grafikprogramme.

[1200px Grid](https://1200px.com/) Aufbauend auf dem 960px Grid, aber 1200px breit. Außerdem in allen Werten anpassbar. **Output:** CSS Dateien oder Photoshop/Illustrator Files.

[Bootstrap](http://getbootstrap.com/2.3.2/index.html) Sehr flexibel und anpassbar, allerdings nur, wenn es im Web verwendet wird, Einbindung in HTML/CSS also Pflicht. **Output:** Im Code verwendbare CSS-Klassen.

[Dead Simple Grid](https://github.com/mourner/dead-simple-grid) Ähnlich wie Bootstrap aber sehr viel simpler. Flexibel anpassbar, aber auch nur für HTML/CSS geeignet. **Output:** Im Code verwendbare CSS-Klassen.

[Bourbon Neat](http://neat.bourbon.io/) Ähnlich wie Bootstrap Dead Simple Grid aber deutlich komplexer, da es zwingen über einen CSS-Präprozessor eingebunden werden muss. Flexibel anpassbar, aber auch nur für HTML/CSS geeignet. **Output:** Im Code verwendbare CSS-Klassen.

[Foundation](http://foundation.zurb.com/) Ähnlich wie Bootstrap. Flexibel anpassbar, aber auch nur für HTML/CSS geeignet. **Output:** Im Code verwendbare CSS-Klassen.

4.2.1 Abbildung im Tool

Da sich die untersuchten Grid Systems in ihren Eigenschaften sehr unterscheiden, sollte das Tool mehrere Empfehlen und ihre typischen Einsatzgebiete nennen. Die Wahl fiel mit gridpack auf ein System, dass auch unabhängig vom Code eingesetzt werden kann sowie auf Foundation, das nur im CSS verwendet werden kann. Foundation wurde gewählt, da es die Möglichkeit

bietet, nur das Grid System ohne Klassen für andere Elemente einzubinden. Um dem Nutzer das Arbeiten mit Grid Systems näher zu bringen, könnte das Tool den Nutzer dazu auffordern, verschiedene Elemente auf einer Seite zu Platzieren. Zunächst sollte die Platzierung ohne eine Hilfestellung stattfinden (s. Abb.X), im zweiten Schritt dann mit einem vom User konfigurierten Grid System (s. Abb.X).

4.3 Grid Systems auf nativen Plattformen

Auf nativen Plattformen werden Grid Systems eher von Interface Designern verwendet, vorgefertigte Grid Systems wie im Web bestehen in der Art nicht. Die Zielgruppe des Tools sind jedoch nicht Interface Designer sondern Studenten, die Projekte in der Rolle des Entwicklers durchführen. Hier müssen also Plattformspezifische Grundlagen gefunden werden.

4.3.1 iOS

Unter iOS spielt das Storyboard in Xcode für die Platzierung von Elementen eine wichtige Rolle. Im Storyboard ist implizit ein Grid vorhanden, dass zumindest die Außenabstände und die Abstände von Elementen untereinander mit Hilfslinien kennzeichnet. (s. Abb. X) In den Human Interface Guidelines werden außerdem die folgenden Abstände definiert: TBD Das Tool sollte an dieser Stelle auf die Werte und die Benutzung des Storyboards hinweisen. Eine weitere Interaktivität macht wenig Sinn.

4.3.2 Android

Für Android gestaltet sich das erstellen von Layouts als schwieriger, da der in Android Studio enthaltene Visuelle Editor deutlich weniger intuitiv ist als bei Xcode. Das Interface wird hier in der Praxis außerdem deutlich häufiger in XML-Form beschreiben, als durch einen visuellen Editor erzeugt. Die Fehleranfälligkeit ist daher besonders hoch. In den [Material Design Guidelines](<https://material.google.com/layout/metrics-keylines.html#metrics-keylines-baseline-grids>) wird generell ein 8dp Square Grid verwendet. Das heißt, der Abstand von allen Elementen

nach außen beträgt mindestens 8dp. Für alle UI-Elemente sind jedoch auch andere Abstände ineinander angegeben, auf die verwiesen werden sollte. Für alle Elemente und Endgeräte sind außerdem Illustrator-Vorlagen vorhanden. Auch hier sollten eher auf die vorhandenen Ressourcen verwiesen werden, um den Umfang des Tools nicht unnötig zu vergrößern.

4.4 Grid Systems im Print

Generell sind Grid Systeme auch im Print-Bereich sehr verbreitet und finden dort sogar ihren Ursprung. Sie unterscheiden sich aber durchaus von Grid-Systemen, wie man sie im Web-Bereich verwendet. Sie haben deutlich weniger Spalten und sind dadurch deutlich weniger fein. In der Regel beschreibt eine Spalte den Platz für den Fließtext, daher sind Auch Raster mit nur einer Spalte sehr verbreitet.

Ein weiteres Beispiel ist der Satzspiegel nennen, der in seiner Funktion auch ein Grid-System ist, jedoch die harmonische Platzierung von Text auf einer Doppelseite beschreibt.

Da viele der im Rahmen der Fehleranalyse untersuchten Artefakte einzelne Textseiten (z.B. Exposés) waren wird der Fokus des Tools darauf liegen, eine einzelne Textseite zu setzen. Hier sind zum einen die Abstände vom Text zum Rand der Seite als auch die Abstände von Spalten untereinander (wenn der Text mehrere Spalten besitzt) interessant.

Weiterhin sollte erwähnt werden, dass die meisten Office-Programme diese Abstände bereits in den Standardeinstellungen auf solide Werte setzen.

4.4.1 Berechnung


Die simpelste mögliche Berechnung verwendet einen Bruchteil der Gesamten Länge der kürzesten Kante des Dokumentes als Außenabstand. Für ein DIN A4 Blatt mit einer Breite von 595px oder 210mm ergibt das folgende Berechnung mit einem Außenabstand vom 10% Kürzeste Kante: 595px bzw. 210mm Ansatz: $595\text{px} \cdot 0.1 = 59,5$ bzw. $210\text{mm} \cdot 0.1 = 21\text{mm}$ Gerundet: 60px bzw. 21mm “ Der Außenabstand würde also 60px bzw. 21mm betragen. Das folgende Bild zeigt ein DIN A4 Dokument mit den errechneten Abständen:

!60px Margin](<https://github.com/Plsr/praxisprojek>

Mit diesem Wert kann also eine Richtlinie angenähert werden. Nach unten lassen sich für den Abstand genaue Grenzen finden: Viele Drucker benötigen einen Druckrand von etwa 15mm, das Tool sollten den Nutzer also warnen, sollte der Abstand nach außen diesen Wert unterschreiten.

Sollte ein layout mit mehr als einer Spalte gewünscht sein, kann der Abstand zwischen ihnen ebenfalls über den errechneten Außenabstand errechnet werden:

“ Errechneter Außenabstand: 60px Innenabstand: 60px / 2 = 30px “

Das folgende Bild zeigt einen zweispaltigen Text mit den errechneten Abständen (außen und innen):  (https://github.com/Plsr/praxisprojekt/blob/master/images/layout/A4

Zu beachten ist hier, dass die Spalten mit diesen Werten nur 445px breit wären und hierdurch du im Kapitel Typographie behandelte optimale Laufweite des Textes von 75 - 90 CPL bei größeren Texten nicht mehr gewährleistet werden kann. Dem Tool sollten die im ersten Schritt errechneten Werte für Texte bekannt sein und es sollte den Nutzer auf diesen Umstand hinweisen.

4.4.2 Abbildung im Tool

Das Tool soll dem User die Möglichkeit geben, ein Format zu wählen. Für dieses Format sollte der User dann die Anzahl der Spalten, die Abstände zwischen den Spalten und die Abstände nach außen einstellen können. Das Tool weist den Nutzer dann darauf hin, welche Werte kritisch sind und warum sie nicht optimal sind. Optional soll noch ein Bild eingefügt werden können. Die Platzierung soll dem Nutzer überlassen werden und ihn dazu bringen, bereits im Tool auch aktiv mit einem Layout zu arbeiten.

Kapitel 5

Einleitung

5.1 Herleitung

Anders als noch bei der Typographie oder beim Layout bewegen wir uns im Bereich der Struktur sehr weit von generell anwendbaren Regeln weg (auch wenn diese schon bei den genannten vorhergehenden Gebieten nicht absolut zu finden sind). Die Fragen, die sich in diesem Kapitel stellen sind also

1. Gibt es überhaupt mathematische Grundlagen, die hier genutzt werden können?
2. Wie werden die Inhalte so vermittelt, dass sie zum Einen universal anwendbar, zum Anderen aber auch so konkret wie möglich sind?

5.2 Whitespace

Das wichtigste Konzept in diesem Zusammenhang ist das des Whitespaces. Weiterhin sollte hier auch das Gesetz der Nähe mit eingebracht werden. Zunächst sollte definiert werden, was Whitespace ist und warum man mit Whitespace arbeiten sollte. Hier ist es schwer, absolute mathematische Regeln zu finden, die man in jedem Fall anwenden kann. Jedoch sollten sich Restriktionen finden lassen, an denen sich der Nutzer später orientieren kann. Die ersten

Einsätze von Whitespace haben wir bereits bei der Typographie erlebt. generell macht ein Vorgehen "von innen nach außen" hier Sinn. So sollten zunächst Abstände von einzelnen Elementen festgelegt werden. Eine Menge dieser Elemente kann dann später zu einer Gruppe zusammengefasst werden (hier kommt das Gesetz der Nähe ins Spiel) und auch dieser Gruppe können wieder Abstände zugewiesen werden. So lässt sich die Struktur der gesamten Seite zumindest in einem gewissen festgelegtem Rahmen bestimmen. Auch hier werden am Ende keine Meisterwerke heraus kommen, aber zumindest eine gewisse Grundqualität in der Struktur kann garantiert werden. Besonderes Augenmerk sollte hier auch darauf gelegt werden, dass die Abstände konsistent sind.

5.3 Was ist Whitespace?

Whitespace (oder auch "negative space") ist leerer Raum. Alex J. White beschreibt ihn wie folgt:

The single most overlooked element in visual design is emptiness. The lack of attention it receives explains the abundance of ugly and unread design. [...] Design elements are always viewed in relation to their surroundings. Emptiness in two-dimensional design is called white space and lies behind the type and imagery. But it is more than just the background of a design, for if a design's background alone were properly constructed, the overall design would immediately double in clarity and usefulness. Thus, when it is used intriguingly, white space becomes foreground. The emptiness becomes a positive shape and the positive and negative areas become intricately linked.

Whitespace ist also kein Raum, den es noch zu füllen gilt sondern bewusst leer gelassener Raum, dessen Aufgabe es ist, Elemente in ihrer Wichtigkeit zu unterstützen. Eine Faustregel ist dabei, dass Elemente, um die herum mehr Whitespace liegt wichtiger sind (vgl. Überschriften in der Typographie). Generell wirken Gestaltungen mit mehr Whitespace auch wertiger. Hauptaugenmerk in diesem Projekt sollte aber darauf liegen, den Whitespace auf ein Mindestmaß zu bringen um eine angenehme visuelle Gestaltung zu erhalten.

Kapitel 6

Einleitung

Neben der Typographie stellte sich das Kapitel Farben als eines der Komplexesten Kapitel heraus. Viele Entscheidungen im Bezug auf Farben müssen auf einer subjektiven Ebene getroffen werden, einige können jedoch, zumindest bedingt, auch durch Regeln vereinfacht werden.

Dieses Kapitel beschäftigt sich mit verschiedenen Fragestellungen. Zunächst wird der Versuch unternommen, herauszufinden, ob die Verwendung bestimmte Farben immer empfohlen oder von dieser abgeraten werden kann. Weiterhin wird versucht, Regeln für die Kombination von verschiedenen Farben zu finden. Die Frage, wie und in welchem Umfang Farben in einer Gestaltung eingesetzt werden sollten wird beantwortet, und es wird der Versuch unternommen, einen Workflow zum finden von Farben zu definieren. Außerdem wird die Verwendung von Farben auf den nativen Plattformen Android und iOS behandelt.

6.1 Farbwirkung

Um herauszufinden, ob bestimmte Farben empfohlen oder von ihnen abgeraten werden kann, soll hier zunächst der psychologische Effekt von Farben auf Menschen behandelt werden.

Generell lässt sich feststellen, dass Farben eine Wirkung auf die menschliche Wahrnehmung und ihr Verhalten haben. In den letzten drei Jahren konnte beispielsweise während des Moduls "Grundlagen der visuellen Kommunikation" beobachtet werden, wie ein Raum von über 100

Menschen fast einstimmig bestimmte Farbkombinationen mit Adjektiven belegten. Auch in der Forschung wird diese Annahme unterstützt:

Color filters humanity's perception of the world and alters people's relationships with their surroundings. It influences human perception, preference, and psychology throughout the lifespan [[Cyr et al.\(2010\)](#) [Cyr, Head, and Larios](#)]

Colour has the potential to elicit emotions or behaviors [...] [[Rider\(2010\)](#)]

Die Frage in diesem Teil lautet also nicht, ob es möglich ist, Menschen mit Farben zu beeinflussen, sondern vielmehr *wie*. Konkreter gilt es die Frage zu beantworten, ob es bestimmte Farben gibt, die bei einem Großteil der Menschen positive bzw. negative Gefühle hervorrufen um so eine solide Basis für das Erstellen eines Farbschemas zu schaffen.

6.2 Beliebtheit von Farben

Generell lässt sich keine absolute und allgemeingültige Regel aufstellen, welche Farben am beliebtesten sind. Dies hängt von verschiedenen Faktoren wie Herkunft, Kultur, Geschlecht und Charakterausprägung des Individuums ab. Für Westeuropäische Erwachsene weist aber die von [[Eysenck\(1941\)](#)] erstellte Hierarchie *blau, rot, grün, violett, orange, gelb* eine hohe Gültigkeit auf.

Es lässt sich aber auch feststellen, dass kalte Farbtöne beliebter sind als warme, was wiederum im Widerspruch mit der oben aufgestellten Hierarchie steht. Zwar scheint Blau immer eine solide Lösung zu sein, jedoch sollte die Farbauswahl auf Basis der generellen Beliebtheit der Farben nur als letztes Mittel erfolgen. Gibt es eine andere Möglichkeit, die Hauptfarbe einer Gestaltung zu definieren (siehe andere Kapitel), so sollte diese auch verwendet werden.

6.3 Psychologische Wirkung von Farben

Eine dieser anderen Möglichkeiten wäre, Farben nach einem intendierten Effekt auf den Betrachter auszuwählen. Hier unterschieden sich sowohl Farbtöne, als auch einzelne Farben. So

erhöhen **warme Farbtöne** wie Rot und Gelb den Puls und verursachen Hunger beim Betrachter [Berman(2010)], einigen **kalte Farbtönen** wie zum Beispiel Blau kann hingegen ein einen beruhigenden Effekt auf den Betrachter nachgewiesen werden [Crozier(1999)].

[Berman(2010)] listet einige der beliebtesten Farben samt ihrer Bedeutung auf. Diese gilt explizit für Nordamerika, sollte aber auch in der restlichen westlichen Welt Gültigkeit haben.

Rot hat, je nach Kontext, verschiedene (und teilweise recht unterschiedliche) Bedeutungen. Sp gilt Rot als Farbe der Romantik, Liebe und Leidenschaft, wird jedoch auch als Signalfarbe auf Straßenschilden genutzt und kann in gewissen Kontexten als Zeichen für Gefahr stehen.

Gelb und Orange rufen Erinnerungen an die Sonne hervor und gelten daher als fröhliche Farben, sind im allgemeinen aber eher unbeliebt. Gelb ist oft heller als Weiß und wird als Signalfarbe genutzt, Orange erhöht (ähnlich wie Rot) den Appetit des Betrachters.

Blau ist in der westlichen Welt die beliebteste Farbe. Ihr wird eine beruhigende Wirkung nachgesagt, in verschiedenen Abstufungen wirkt Blau jedoch verschieden, von dynamisch bis zuverlässig.

Grün steht hauptsächlich als Farbe der Natur und der Gesundheit, kann in dunkleren Abstufungen aber auch für Wohlstand stehen.

Violett steht für das Noble und Majestätische, kann gleichzeitig aber auch für Einsamkeit stehen.

Es ist also schwer, einer Farbe genau eine Eigenschaft zuzuschreiben, häufig hat die gleiche Farbe in einer helleren oder dunkleren Abstufung eine andere Wirkung. Zumindest lässt sich aber eine grobe Auflistung von Eigenschaften finden, auf dessen Basis eine Farbempfehlung erfolgen könnte.

6.4 Kontraste

Um eine Farbpalette zu erstellen müssen Farben kombiniert werden. Um diese Farben so zu kombinieren, dass sie zusammen funktionieren, gibt es verschiedene Farbkontraste, mit denen gearbeitet werden kann. Dieses Kapitel beschäftigt sich mit der Frage, welche Kontraste es

gibt, welche davon am Besten programmatisch umzusetzen sind und welche letzten endes im Tool verwendet werden sollten.

Weiterhin werden einige bereits bestehende Tools auf ihre Stärken und Schwächen untersucht.

Zu Farbschemata, deren Verwendung und der Anzahl der vertretenen Farben schreibt Rose Rider:

Strong color schemes are important to effective design. Monochromatic designs, containing various shades and tints of one hue, are most effective for communicating simple messages (Fehrman & Fehrman, 2004). Multiple-color combinations can add depth, complexity, and additional meaning to design. A common two-hue combination involves complimentary colors, colors directly opposite each other on the color wheel (e.g., blue and orange, or red and green; Berman, 2007). Three-hue color harmonies often involve triads, colors located equally far from each other on the color wheel (Bleicher, 2005).

und

Four-color schemes in general are more attractive than one- or two-color schemes (Shank & LaGarce, 1990). One-color schemes, however, were ranked more tasteful than their two- or four-color counterparts. A simple monochromatic color scheme can lend an air of sophisticated restraint while cutting costs (Nelson, 1994).

Die Frage nach der richtigen Anzahl von Farben in einem Farbschema kann also nicht eindeutig beantwortet werden. Das Tool sollte aber ein Farbschemata mit maximal vier Farben empfehlen, nach Möglichkeit jedoch weniger, um die Zahl der Fehlerquellen möglichst gering zu halten.

In "Kunst der Farbe" beschreibt [\[Itten\(2010\)\]](#) 7 Farbkontraste:

1. Farbe-an-sich-Kontrast
2. Hell-Dunkel-Kontrast

3. Kalt-Warm-Kontrast
4. Komplementär-Kontrast
5. Simultan-Kontrast
6. Qualitäts-Kontrast
7. Quantitäts-Kontrast

Für den hier gegebenen Kontext wurden aus dieser Liste vier Kontraste ausgewählt, die zum Einen in vielen der untersuchten Tools verwendet wurden und zum Anderen einfach umzusetzen sind. Diese sollen im Folgenden kurz erläutert werden.

Der **Kalt-Warm-Kontrast** beschreibt eine Verwendung von Kalten und Warmen Farben im Farbschema. Itten unterteilt den Farbkreis dabei wie folgt:

Die Farben Gelb, Orange, Rotorange, Rot und Rotviolett werden im allgemeinen als warme, und Gelbgrün, Grün, Blaugrün, Blau, Blauviolett und Violett werden als kalte Farben bezeichnet. [Itten(2010), S. 45]

Zwar wurde die Verwendung diese Kontraste in den untersuchten Tools nicht explizit gefunden, jedoch liegt er wegen der unternommenen Teilung des Farbkreises nahe am Komplementär-Kontrast (So werden häufig Blau und Orange in Farbschemata verwendet).

Der **Komplementär-Kontrast** ist der am häufigsten in bestehenden Lösungen verwendete Kontrast, vermutlich wegen seiner einfachen logischen Umsetzung. Zwei Farben gelten als Komplementär, wenn sie zusammengemischt Grau ergeben. Da das Auge auch selbstständig nach einer Komplementärfarbe sucht, wenn diese nicht gegeben ist [Itten(2010), S. 49], wirkt das auftreten zweier Komplementärfarben in einem Farbschema sehr harmonisch.

Der **Qualitäts-Kontrast** ist, ähnlich wie der Komplementär-Kontrast sehr einfach zu erzeugen. Als Qualitätskontrast bezeichnet Itten den Gegensatz von gesättigten und stumpfen Farben [Itten(2010), S. 55]. Dieser lässt sich also sehr einfach durch das Mischen einer Grundfarbe mit Schwarz, Weiss, Grau oder der entsprechenden Komplementärfarbe.

Der **Quantitäts-Kontrast** ist für das Finden einer Farbpalette weniger von Bedeutung als für ihren späteren Einsatz. Dieser Kontrast wird also im Kapitel "Farbeinsatz" noch einmal von Relevanz sein, trotzdem sollte er hier kurz erläutert werden. Der Quantitäts-Kontrast bezieht sich auf das Größenverhältnis zwischen zwei verwendeten Farben, also "viel und wenig" oder "groß und klein" [Itten(2010), S. 59]

6.4.1 Bestehende Lösungen zur Farbfindung

Im Rahmen der Recherche wurde einige bestehende Lösungen untersucht. Die folgende Liste gibt einen kurzen Überblick über einige der untersuchten Tools und deren Lösungsansätze.

Paletton Sehr simpel zu benutzen, der Nutzer wählt eine Grundfarbe und kann zwischen Benachbarten, Triadisch oder Tetraedisch angelegten Farben wählen. Dabei besteht immer auch die Möglichkeit, die Komplementärfarbe mit in die Farbpalette zu übernehmen. Das Tool erzeugt eine Farbpalette von 2 bis 4 Farben.

Adobe Color CC Etwas umfangreicher als Paletton, es besteht die Wahl zwischen 6 verschiedenen Kontrasten. Das Tool erstellt eine Farbpalette von 5 Farben, die alle individuell angepasst werden können.

Coolors Sehr einfach zu bedienen, der Nutzer drückt die Leertaste und es wird eine neue Palette erzeugt. Farben, die der Nutzer mag können "gelockt" werden und bleiben bei der nächsten Generierung erhalten. Es wird für den Nutzer nicht deutlich, auf welcher Grundlage die Farben ausgesucht werden. Das Tool liefert eine Farbpalette von 5 Farben.

colourlovers Liefert beim Erstellen einer Farbpalette nur wenig Hilfe (die einzige Hilfe ist das Erstellen einer Palette aus einem Foto). Auf der Plattform sind von anderen Nutzern erstellte Farbpaletten vorhanden, die zur Inspiration dienen können.

color-scheme-js JavaScript Library, die Farbpaletten erstellt. Es gibt 5 verschiedene Variationen, ähnlich wie bei Adobe Color CC. Ein Beispiel kann [hier](http://c0bra.github.io/color-scheme-js/) eingesehen werden. Leider liegt keine Lizenz bei, es gilt also zu prüfen, ob diese Library verwendet werden könnte.

Viele der untersuchten Tools liefern gute und verwendbare Ergebnisse, jedoch vermittelt keines Interaktiv Wissen, dem Benutzer wird lediglich eine Aufgabe abgenommen. In diesem Punkt könnte die in diesem Projekt zu entwickelnde Lösung ansetzen.

6.5 Einsatz von Farben

Nachdem nun einige Methodiken zum Finden von Farben besprochen wurden gilt es weiterhin die Frage zu klären, wie diese Farben einzusetzen sind. So stellt sich zum Beispiel die Frage, ob es einen prozentualen Richtwert dafür gibt, welche Elemente in einer Gestaltung farbig sein sollten oder ob sich bestimmte Elemente finden lassen, die immer farbig sein sollten.

Generell lassen sich für den Anteil der Farbigen Elemente, etwa auf prozentualer Basis, keine Regeln formulieren. Zwar zeigen Studien, dass Webseiten mit wenigen Farben in der Regel als hochwertiger und teurer angesehen werden [?], jedoch bezieht sich diese Beobachtung nicht darauf, ob eine Gestaltung als angenehm wahrgenommen wird oder nicht. Gerade im Kontext eines Informatikstudienganges können auch bunte Gestaltungen durchaus reizvoll und angebracht sein.

Für den Anteil der Farbigen Element in sich lassen sich jedoch zumindest Ansätze von finden. So führt [Itten(2010), S. 59] zu den bereits früher erwähnten Quantitäts-Kontrasten Verhältnisse der Grundfarben zueinander auf:

Gelb : Orange : Rot : Violett : Blau : Grün sind wie 3 : 4 : 6 : 9 : 8 : 6

Verwendet der Benutzer also zwei oder mehr dieser Farben, kann zumindest für deren Einsatz eine Empfehlung gegeben werden.

Weiterhin lässt sich feststellen, dass die für die intendierte Botschaft der Gestaltung besonders wichtigen Elemente häufig farbig sind. Hierzu können zum Beispiel ein Button zählen, den der User drücken soll oder ein besonders wichtiges Zitat, das die Grundaussage des gesamten Textes gut widerspiegelt.

6.6 Farbfindung

Nachdem die vorhergehenden Abschnitte einige theoretische Grundlagen lieferten beschäftigt sich dieses Kapitel mit der konkreten Erstellung von Farbpaletten. Dieser Prozess wird hier in drei Schritte unterteilt und soll so auch im Tool abgebildet werden. Den ersten Schritt stellt das Finden einer Grundfarbe dar, auf der aufbauend dann die anderen Farben der Palette gefunden werden können. Im zweiten Schritt gilt es, je nach Art der Farbpalette, entweder einer Akzentfarbe oder passende Abstufungen der Grundfarbe zu finden. Im letzten Schritt soll die Farbpalette dann mit passenden Grautönen vervollständigt werden. Für das Finden der passenden Grautöne soll sich das Tool am Vorgehen orientieren, das [?] beschreibt.

6.6.1 Finden einer Grundfarbe

Das Finden einer Grundfarbe wurde bereits im Abschnitt Farbwirkung angeschnitten. In vielen Fällen ist die Grundfarbe bereits vorgegeben oder es gibt Anhaltspunkte, an denen diese definiert werden kann. Solch ein Anhaltspunkt könnte beispielsweise ein Logo sein. Oft liegen Vorlagen aus den entsprechenden Modulen vor, an denen man sich orientieren kann (und sollte). Außerdem sind Styleguides sowohl von der Technischen Hochschule Köln als auch der Medieninformatik vorhanden. Diese enthalten zwar bereits vordefinierte Farbpaletten und machen diesen Schritt im Tool überflüssig, sollten aber gegenüber der individuellen Definition einer Farbpalette den Vorzug erhalten. Weiterhin gibt es für native Plattformen bereits Farbvorgaben, siehe hierzu das entsprechende Kapitel.

Lässt sich all das oben genannte nicht anwenden, so muss eine andere Grundfarbe gefunden werden. Wie bereits im Abschnitt Farbwirkung erwähnt gibt es hier keine universellen Regeln, auf deren Grundlage man eine sichere Entscheidung treffen könnte, hier spielen viel Charakteristiken der Zielgruppe eine Rolle. Die Zielgruppe für Artefakte kann mit recht großer Sicherheit als in Westeuropa lebende Erwachsene definiert werden, wodurch sich konkretere Regeln entwickeln lassen. Der erste Schritt wäre die Auswahl eines Farbtones, hier kommt es ganz darauf an, ob die Anwendung eher ruhig oder eher aufregend wirken soll. Generell scheint sich aufgrund der Beliebtheit der Farbe auch das Verwenden eines Blautones empfehlen. Auch finden

sich im Web viele Listen mit Adjektiven, die bestimmten Farben zugeschrieben werden. Auch diese Listen wäre ein valider Ansatzpunkt.

Als erster Schritt im Tool wäre es also denkbar, dem Nutzer aufzuzeigen, wo er eine vielleicht bereits definierte Grundfarbe finden könnte und diese über einen simplen Colorpicker einzulesen. Außerdem wäre es möglich, dem Nutzer eine Liste von Adjektiven zur intendierten Wirkung seiner Gestaltung zu präsentieren und aufgrund der Auswahl des Nutzer eine Farbe vorzuschlagen.

6.6.2 Finden einer Akzentfarbe

Für die Akzentfarbe ist vor allem wichtig, dass sie mit der Grundfarbe harmoniert. Außerdem sollte sich deutlich von der Grundfarbe unterscheiden. (Die Ausnahme bildet hier ein Monochromatisches Farbschema, bei dem es keine Akzentfarbe an sich gibt.) Je nach Farbschema gibt es nicht nur eine Akzentfarbe sondern auch mehrere. Zum finden der Akzentfarbe bietet sich zunächst die Komplementärfarbe der Grundfarbe an. Da diese auf der anderen Seite des Farbkreises liegt ist sichergestellt, dass sie sich von der Grundfarbe abhebt (also einen Akzent setzen kann) und mit ihr harmoniert. Sollte mehr als eine Akzentfarbe benötigt werden kann hier auf die triadische oder tetradische Farbfindung zurück gegriffen werden.

Für die tatsächliche Berechnung im Tool müssen die verwendeten Farben in den HSL-Farbraum (Hue, Saturation, Lightness) konvertiert werden. In diesem Farbraum werden Farbtöne radial auf einem Zylinder angeordnet [?].

Um die Komplementärfarbe einer gegebenen Grundfarbe zu finden muss also nur die auf dem Kreis gegenüberliegende Farbe gefunden werden, der Winkel also um 180° vergrößert werden. Das vorgehen beim Finden von 3 oder 4 Farben ist simultan, der Winkel wird hier entsprechend um 120° oder 90° vergrößert.

st die Grundfarbe zum Beispiel ein Rot mit dem HEX-Wert #ff0000 würde dieses zunächst in den HSL-Farbraum konvertiert und die Werte $H = 0^\circ$, $S = 100\%$ und $L = 50\%$ ergeben. Saturation und Lightness werden beibehalten, der Hue-Wert wird um 180° erhöht. Die Komplementärfarbe hat also die Werte $H = 180^\circ$, $S = 100\%$ und $L = 50\%$ oder #00ffff.

Für eine Abstufung von Farben zur Verwendung in einem Monochromatischen Farbschema kann der Hue-Wert der Grundfarbe beibehalten werden und lediglich die Saturation oder Lightness erhöht bzw. verringert werden. Hier sollten dem User 4 zusätzliche Farben in verschiedenen Abstufungen vorgeschlagen werden.

6.6.3 Komplettieren der Farbpalette

Nachdem Grund- und Akzentfarbe gefunden sind fehlen für eine nutzbare Farbpalette noch Grautöne. Hier sollte darauf geachtet werden, nicht zu viele Grautöne zu verwenden, die sich nur marginal unterscheiden. In den meisten Fällen sollten 2-3 Grautöne in verschiedenen Abstufungen ausreichend sein. Als einfachste Methode würde es sich anbieten, neutrale Grautöne ohne andere Farben zu wählen. Welche Grautöne genau genutzt werden muss dabei Subjektiv entschieden werden. Denkbar wären zum Beispiel #eeeeee und #666666.

[?] erläutert eine andere Methode: Bei dieser Methode wird die Grundfarbe mit in die Grautöne eingearbeitet, was ein harmonischeres Bild erzeugt. Für die Berechnung dieser Werte muss jedoch auf eine externe Bibliothek wie zum Beispiel `chroma.js` zurück gegriffen werden.

6.6.4 Signalfarben

Die üblichen Signalfarben fallen aus der Farbpalette heraus. Diese sind in der Regel Grün, Gelb und Rot. Dies bedeutet nicht, dass Grün, Gelb und Rot nicht als Farben in der Farbpalette vorkommen können, sie spielen einfach eine zusätzliche Rolle. Die Farben für die Buttons können zum Beispiel vom Framework Bootstrap übernommen werden. Diese sind:

- Grün: #5cb85c
- Gelb: #f0ad4e
- Rot: #c9302c

Hier stellt sich außerdem die Frage, wie häufig diese Warnfarben tatsächlich benötigt werden. Hauptsächlich treten sie in interaktiven Systemen auf, viele dieser interaktiven Systeme werden

aber in Modulen an der Hochschule in Android umgesetzt, welches seine eigenen Guidelines zu Farben mitbringt.

6.7 Android & iOS

Für beide Betriebssysteme liegen Guidelines bezüglich der visuellen Gestaltung, auch im Bezug auf Farben vor. An diese soll sich auch das Tool halten. Das Tool soll hier aber nicht nur auf diese Guidelines verweisen oder sie wiederholen, sondern eine interaktive Möglichkeit bilden, auch für diese Plattformen eine Farbpalette zu erstellen. Da die beiden Plattformen sich in ihren Richtlinien deutlich unterscheiden werden sie hier auch getrennt behandelt.

6.7.1 Android

Die Android Guidelines geben keine spezifischen Farben vor, das Hauptaugenmerk liegt darauf, dass interaktive Elemente genügend Kontrast aufweisen.

Your app's color palette may be defined by using a custom palette suited to your brand, [...]. Alternatively, you may use the material design color palette. All color palettes should include sufficient contrast between different UI elements.

Da für Projekte im Hochschulkontext davon ausgegangen werden kann, dass keine Brand Identity vorliegt soll der Fokus hier darauf liegen, die von Google vordefinierten Farben zu verwenden. Vor allem soll interaktiv aufgezeigt werden, welche Elemente wie gefärbt werden sollten.

Die Android Guidelines geben eine Menge an Farben vor. Zusätzlich sind alle Farben mit einem Indikator für die Helligkeit versehen. Die Guidelines empfehlen die Farben mit der Helligkeit 500 als Grundfarbe zu benutzen. Hiervon bilden sich dann hellere und dunklere Varianten, die für andere Elemente verwendet werden. Die Guidelines stellen außerdem eine Menge von Akzentfarben bereit, die frei mit einer der Grundfarben kombiniert werden können. Die Akzentfarbe sollte für interaktive Elemente verwendet werden.

6.7.2 iOS

Die iOS-Richtlinien sind deutlich weniger konkret. Unter iOS wird Farbe vor allem dafür genutzt, um Interaktivität deutlich zu machen. Hier werden als Referenz die Systemfarben genannt.

In iOS, color can indicate interactivity, impart vitality, and provide visual continuity. Look to the system's color scheme for guidance when picking app tint colors that look great individually and in combination, on both light and dark backgrounds.

Hier wäre sogar das Vorschlagen nur einer einzigen Farbe eine valide Lösung.

Kapitel 7

Einleitung

Der folgende Abschnitt bezieht sich auf den technischen Einsatz von Bildern in einer Gestaltung. Themen wie Ästhetik oder das Finden von passenden Bildern für eine Gestaltung werden hier nicht behandelt. Diese Themenabgrenzung begründet sich zum einen mit dem schieren Umfang des Themas Ästhetik und Fotografie, der im Rahmen dieses Projektes nicht abgedeckt werden kann. Zum Anderen ist die Abgrenzung eine Folge der Fehleranalyse, in der vor allem technische Mängel auffielen. Weiterhin ist es gerade in einem Wissenschaftlichen Studiengang nicht immer möglich, ästhetisch schöne Bilder zu verwenden.

Während der Fehleranalyse fielen vor allem zwei Arten von Fehlern auf: Strecken bzw. stauchen von Bildern und das verwenden von verpixelten Bildern.

Im Folgenden sollen die Hauptursachen für diese Fehler definiert und Ansätze entwickelt werden, wie diese vermieden werden können. Außerdem gilt es festzustellen, ob, und wenn ja, wie die erarbeiteten Techniken im Tool möglichst interaktiv vermittelt werden können.

7.1 Strecken & Stauchen

Gestreckte und gestauchte Bilder in einer Gestaltung sorgen beim Betrachter für einen unprofessionellen Eindruck. Zwar konnten keine wissenschaftlichen Belege dafür gefunden werden, aber welchem Grad der Verzerrung Menschen ein Bild als verzerrt erkennen und wie schnell,

jedoch kann mit einiger Sicherheit davon ausgegangen werden, dass die meisten Menschen ein verzerrtes Bild erkennen, gerade, wenn sich im Bild ihnen bekannte Gegenstände befinden. Ursache für ein Strecken oder Stauchen des Bildes ist dabei in den meisten Fällen das Nichteinhalten des ursprünglichen Seitenverhältnisses beim skalieren des Bildes.

Als Seitenverhältnis wird das Verhältnis der beiden Kanten eines Bildes zueinander bezeichnet. Obwohl die Berechnung des Seitenverhältnisses in der Praxis eher unwichtig ist (in der Regel errechnen Programme dieses Verhältnis automatisch) kann sie für das Tool dennoch wichtig werden, um eine korrekte Skalierung zu überprüfen. Wie bereits durch die Schreibweise angedeutet ('4:3', '16:9') lässt sich das Seitenverhältnis berechnen durch

$$\text{Seitenverhältnis} = \text{Breite des Bildes} / \text{Höhe des Bildes}$$

Für ein Bild mit den Maßen 400px * 300px ergibt sich also ein Seitenverhältnis von 1,33 oder 4:3. Der Schritt von der Dezimalzahl zur "schönen" Darstellung kann dabei im Tool vernachlässigt werden, da die Berechnung nur intern erfolgt.

Beim Skalieren des Bildes sollte also darauf geachtet werden, beide Kanten im gleichen Verhältnis zu skalieren (s. Abb. 1 & 2), um das Bild nicht zu strecken oder zu stauchen (s. Abb. 3)

7.1.1 Konkrete Anwendung

In den meisten Textverarbeitungsprogrammen sowie Grafikprogrammen und Editoren für native Plattformen ist die Anwendung denkbar simpel: Bei der Bearbeitung eines Bildes werden verschiedene Punkte angeboten, mit denen das Bild verändert werden kann. Hier gilt es, einen der Eckpunkte zu verwenden, um das Seitenverhältnis des Original-Bildes beizubehalten (s. Abb. 4)

Auch im Web ist die richtige Skalierung von Bildern leicht zu erreichen. Die CSS-Attribute width und height haben einen Standardwert von auto [http://www.w3schools.com/css/css_dimension.asp], alle Bilder werden also zunächst

HTML:

```

```

CSS:

```
img {  
  width: 400px;  
  height: auto;  
}
```

7.1.2 Eine gegebene Fläche ausfüllen

Häufig tritt eine falsche Formatierung vor allem dann auf, wenn das Bild eine Fläche füllen soll, die nicht dem Seitenverhältnis des Originalbildes entspricht (Abb. 5 illustriert das Problem).

Ein reines Skalieren des Bildes reicht hier nicht aus, um ein befriedigendes Ergebnis zu erzielen. Im zweiten Schritt muss das Bild außerdem beschnitten werden (s. Abb. 6).

Welcher Teil des Bildes abgeschnitten wird und welcher erhalten bleibt ist dabei wiederum eine ästhetische Frage und kann nur subjektiv und je nach Situation beurteilt werden. Optimaler Weise wird die Kernaussage des Bildes durch die Beschneidung nicht verändert. Das Tool könnte dies beispielhaft mit einigen wenigen vordefinierten Bildern abbilden.

Grafikprogramme bieten für das behandeln dieser Aufgabe verschiedenen Möglichkeiten, sowie das einfach beschneiden des Bildes oder den Einsatz von Masken.

Auch im Web bieten sich verschiedene Möglichkeiten an. Die meiste Kontrolle über die Ausrichtung des beschnittenen Bildes bringt dabei die Verwendung von background-image mit sich:

HTML:

```
<div class="bg-image"></div>
```

CSS:

```
.bg-image {  
  width: 300px;  
  height: 400px;  
  background-image: url("image/landscape.jpg");  
  background-size: cover;  
}
```

7.2 Verpixelte Bilder

Neben gestreckten und gestauchten Bildern fallen auch verpixelte Bilder unschön auf und lassen die Qualität der Gestaltung schlechter wirken. Die Ursachen für verpixelte Bilder können verschieden sein.

Die Ursache, die am simpelsten vermieden werden kann, ist das Skalieren von Bildern über ihre Originalgröße hinweg. Eine weitere Ursache kann in der Pixeldichte des Mediums liegen, auf dem die Bilder betrachtet werden. Beide Probleme treten nur bei der Verwendung von Bitmaps auf.

7.2.1 Bitmaps

Enderle, Kansy und Pfaff definieren Bitmaps oder Raster Graphics wie folgt:

Raster graphics — Definition Computer graphics in which a display image is composed of an array of pixels arranged in rows and columns. [Enderle, Gnter, Klaus Kansy, and Gnther Pfaff. *Computer Graphics Programming: G*

Bitmaps bestehen also aus einem Array von Pixeln, von denen jeder seine eigene Farbe und Position hat (s. Abb. 8).

Beim skalieren dieses Bildes muss also eine neue Anzahl an Pixeln innerhalb dieses Rasters erzeugt werden. Beim Skalieren nach unten ist dies kein Problem, da benachbarte Pixel zu einem neuen "verschmelzen" und aufgrund des kleineren Bildes kein Unterschied zu erkennen ist. Beim skalieren nach oben müssen aber neue Pixel aus der Datengrundlage des Bildes erzeugt werden. Bei der Skalierung können zwar neue Pixel anhand der vorhandenen, ähnlich wie bei der Skalierung nach unten, errechnet werden, jedoch fällt die fehlende Grundlage zur Berechnung schnell auf und das Bild wirkt verschwommen.

Nach Möglichkeit sollten Bilder also nicht größer skaliert werden als ihre Originalgröße zulässt.

7.3 PPI & DPI

Die Begriffe PPI (Pixels per Inch) und DPI (Dots per Inch) werden insbesondere in zwei Bereichen wichtig: * Hochauflösende Displays (z.B. Retina) * Print

Die Begriffe bezeichnen die Pixel bzw. Dots, die ein Ausgabegerät (z.B. Display oder Drucker) pro Inch darstellen kann. Zu Problemen, besonders im Bezug auf Bilder kommt es hier, wenn die Pixeldichte der Bilder nicht mit der des Ausgabegerätes übereinstimmt. Hier erfolgt dann eine Skalierung nach oben, die schon im vorherigen Kapitel besprochen wurde.

Das Tool soll hier keine Schritt für Schritt Anleitung geben, wie hier zu verfahren ist, sondern eher eine erinnernde Funktion übernehmen und den Nutzer darauf hinweisen, Bilder unter Umständen in die richtige Pixeldichte konvertiert werden müssen.

Kapitel 8

Einleitung

Das nachfolgende Kapitel beschäftigt sich mit der Darstellung von interaktiven Elementen. Dieser Bereich der visuellen Gestaltung weist starke Überschneidungen mit den Themengebieten der Mensch-Computer-Interaktion auf. Diese sollen hier nur so grundlegend wie irgend möglich behandelt werden, um nicht zu sehr vom gesetzten Ziel des Endproduktes abzuweichen und den Focus auf der visuellen Gestaltung zu belassen.

Zunächst sollte hier jedoch der Begriff "interaktive Elemente" erläutert werden. Als interaktives Element wird im folgenden jedes Element verstanden, mit dem der Nutzer interagieren kann und in Folge dessen sich der Zustand der Anwendung verändert. Interaktive Elemente kommen also nur in den Interaktiven Artefakten vor und entfallen in allen untersuchten textbasierten Medien.

Während der Fehleranalyse wurde vor allem deutlich, dass interaktive Elemente nicht als solche gekennzeichnet wurden, der Nutzer also nicht intuitiv erkennen konnte, mit welchen Elementen er interagieren konnte. Weiterhin wurden interaktive Elemente häufig nicht an einen gegebenen Nutzungskontext angepasst (so erfordern mobile Endgeräte mit Touch-Interface eine andere Gestaltung der Elemente als ein Desktop-Computer mit einer Maus).

Dieses Kapitel gliedert sich in zwei Teile: Zunächst sollen einige generelle Richtlinien im Bezug auf interaktive Elemente festgehalten werden. Im zweiten Teil wird dann auf die Besonderheiten von verschiedenen interaktiven Elementen und auf verschiedenen Plattformen eingegangen.

Generell zu beachten Laut den von Dix angesprochenen *Universal Design Principles*, sollte ein interaktives System (Dix, A. (2004). *Human-computer interaction*. Harlow, England New York : Pearson/Prentice-Hall, P.367)* Dies gilt somit auch und vor allem für die Elemente in diesem System, mit dem der User interagieren kann.

Für die meisten interaktiven Elemente gibt es einige Grundregeln und best practices, die sich in jedem dieser Elemente wiederfinden. Darauf aufbauen werden sie dann für die aktuelle Gestaltung angepasst (beispielsweise durch die Anpassung der Farbe oder die Änderung der Grundform passend zur aktuellen Gestaltung). Hier sollte das Tool auf die bestehenden Regeln zurück greifen. Je nach Element soll im Folgenden validiert werden, ob und in welchem Maße eine Anpassung der Elemente empfehlenswert ist.

Ein weiterer Faktor neben dem Feedback, der sich auf die Benutzbarkeit von interaktiven Elementen auswirkt, ist ihre Größe. Hier finden sich viele theoretische Grundlagen in der Mensch-Computer-Interaktion, letztenendes wird die Größe der Elemente jedoch in der visuellen Gestaltung festgelegt. Auch wenn das Tool seinen Fokus darauf legt, eine gute visuelle Grundlage zu schaffen sollte eine grundlegende Nutzerfreundlichkeit nicht außer Acht gelassen werden.

Die Größe von interaktiven Elementen spielt vor allem auf Endgeräten mit einem Touch-Interface eine große Rolle. Während sich mit der Maus auch kleine Bereiche gezielt auswählen lassen ist dies mit einem Finger nicht ohne weiteres möglich. Eine Studie von Park, Han, Park, Cho belegt, dass die Fehlerrate bei kleineren Elementen deutlich ansteigt:

Small touch keys have poor performance in terms of the success rate and the number of errors according to the results.

Für die Größe von Touch-Elementen gibt es verschiedene Richtlinien von Unternehmen, deren Produktreihe Touch-Devices umfasst: [Apple](<https://developer.apple.com/ios/human-interface-guidelines/visual-design/layout/>) empfiehlt einen Touch-Bereich von mindestens 44pt mal 44pt, [Google](<https://developers.google.com/speed/docs/insights/SizeTapTargetsAppropriately>) spricht von 48px im CSS und [Microsoft](<https://msdn.microsoft.com/en-us/windows/uwp/input-and-devices/guidelines-for-targeting>) empfiehlt 40px. Zwar gibt es keinen definierten Standard für diese Größe, jedoch scheint eine Empfehlung im Bereich von 40px bis 50px valide zu sein.

Elemente Dieser Abschnitt beschäftigt sich mit einer Auswahl verschiedener interaktiver Elemente und ihrer Besonderheiten im Bezug auf Feedback und Größe. Je nach Element wird

hier auch auf Besonderheiten auf den verschiedenen Plattformen iOS, Android und Web eingegangen.

8.0.1 Buttons

Wie viele der interaktiven Elemente stellen Buttons eine Metapher dar:

These regions are referred to as buttons because they are purposely made to resemble the push buttons you would find on control panel.

Wie Knöpfe an einem Kontrollpult führen Buttons dabei eine Änderung des Zustandes der Applikation herbei, beispielsweise durch bestätigen einer Interaktion oder absenden eines Formulars.

Buttons sollten eine visuell klar abgetrennte Region im Interface bilden. Buttons führen nur eine Aktion aus, diese ist durch einen Text und/oder ein Icon repräsentiert (auch hier lassen sich viele Diskussionen im Bereich der MCI darüber finden, in welcher Form die Aktion am Besten kommuniziert werden sollte.) Klassische Button bestehen also aus einem Vordergrund mit Text und/oder Icon und einem Hintergrund und reagieren auf Interaktionen wie z.B. das darüber fahren mit der Maus oder das “drücken” des Buttons. In manchen Betriebssystemen (z.B. den neuere Versionen von iOS) können Buttons auch als reiner Text auftreten. Der Hintergrund ist dann nicht mehr sichtbar, jedoch bleibt die Click-Area gleich. Weiterhin gibt es Versionen, bei denen der Hintergrund transparent ist und nur die Abgrenzungen des Buttons markiert werden. Es bietet sich an, Buttons (und auch andere Elemente) mit der Akzentfarbe zu kennzeichnen, um sie aus der Gestaltung heraus stehen zu lassen.

8.0.1.1 Web

Im Web werden Buttons verschieden definiert. Zum einen gibt es ein explizites ‘<button>’ element, was einen Button im eigentlichen Sinne darstellt und zum Beispiel zum absenden eines Formulars verwendet wird (‘<input style=“submit”>’ ist eine andere, aber ähnliche Variante).

Diese Elemente sind bereits ohne Verwendung von jeglichem CSS wie Buttons gestaltet. Die Gestaltung ist dabei eher neutral und hängt vom verwendeten Betriebssystem und Browser ab.

Zum anderen werden anchor tags häufig visuell wie Buttons gestaltet. Diese haben keinen visuellen Fallback und müssen manuell wie Buttons gestaltet werden.

8.0.1.2 Android

Die Material Design Guidelines spezifizieren 3 unterschiedliche Arten von Buttons: 1. Floating Action Buttons 2. Raised Button 3. Flat Button

Da die Buttons in ihrer Benutzbarkeit bereits sehr durchdacht sind sollen diese im Folgenden nur kurz erläutert werden.

[FloatingActionButton](<https://material.google.com/components/buttons-floating-action-button.html>) pro Screen gibt es maximal einen FloatingActionButton. Der FloatingActionButton repräsentiert die

Mit den vom User zuvor ausgewählten Farben können der Normale und der Gedrückte Zustand des Buttons sehr simpel dargestellt werden. Der Schatten wird hier nicht mit dargestellt, da es für diesen eine Funktion zur Berechnung in der Android Library gibt und genaue Daten nicht bekannt sind. Das nachfolgende Bild zeigt die Färbung am Beispiel der Akzentfarbe Pink (A200) auf. Im gepressten Zustand würde der Button entsprechend in einer dunkleren Abstufung (A400) gefärbt.

[RaisedButtons](<https://material.google.com/components/buttons.html#buttons-raised-buttons>) RaisedButton sollten genutzt werden, um Interaktivität auf einem Screen deutlich zu machen. Sie sollten einen Radius von 2dp und ihre Beschriftung immer groß geschrieben. Ihr Verhalten bei Interaktion ist ähnlich dem von FlatButtons.

[FlatButtons](<https://material.google.com/components/buttons.html#buttons-flat-buttons>) FlatButtons sollten verwendet werden, wenn der Button nur als visueller Hinweis dienen soll. Ihre Höhe: 36dp (48dp touch target) * Breite: mindestens 88dp * Horizontaler Innenabstand: 8dp * Horizontaler Außenabstand: 8dp FlatButtons werden in der Regel in der Grundfarbe dargestellt.

8.0.1.3 iOS

Die Human Interface Guidelines sind im Bezug auf die visuelle Gestaltung von Buttons deutlich weniger explizit als die Material Design Guidelines und beziehen sich eher auf die Verwendung der verschiedenen Button-Typen. Die Guidelines lassen es dem Entwickler offen, komplett eigene Buttons zu gestalten:

You can also design fully custom buttons.

Hier soll trotzdem zum Einen auf die System Buttons und zum anderen auf die anderen vorgestellten Buttons eingegangen werden. Generell sollte auch hier als Farbe für die Buttons die Akzentfarbe verwendet werden. Die Größe der Buttons wird im Xcode Storyboard automatisch gesetzt und sollte so auch beibehalten werden.

systemButtons, in der Regel haben System Buttons weder einen Rahmen noch einen Hintergrund und bestehen lediglich aus Text. Die Guidelines spezifizieren noch drei weitere Typen von Buttons, die sich zwar in ihrem Einsatz unterscheiden, aber in der Gestaltung sehr ähnlich sind.

8.0.2 Input Fields

Viele der Ansprüche an Buttons treffen auch auf Input Fields zu. Zwar dienen sie dem annehmen von User Input, trotzdem sollten sie ein ausreichend großes Touch Target darstellen. Außerdem sollte klar gekennzeichnet sein, wenn ein Input-Field aktiv ist oder wenn in ihm ein Fehler gefunden wurde.

Zwar existieren je nach Datenart auch verschiedene Input Fields mit individuellen Unterschieden im Bezug auf ihre Darstellung, jedoch soll hier stellvertretend nur das Text-Field angesprochen werden, da sich die meisten der Eigenschaften zwischen den verschiedenen Input-Fields problemlos übertragen lassen. Das Tool sollte dem Benutzer jedoch nahelegen, die den Eingabedaten entsprechenden Input-Fields zu verwenden.

8.0.2.1 Web

Im Web wird ein Text Field über ein 'input'-Element mit dem type 'text' erstellt (simultan kann 'text' mit anderen Datenarten wie zum Beispiel 'password' ersetzt werden). Eine Simple Umsetzung eines Text Fields mit erklärendem Platzhaltertext sähe wie folgt aus:

```
<input name="firstname" placeholder="First name" type="text"></input>
```

Alternativ kann statt dem Platzhaltertext auch ein Label außerhalb des input fields vergeben werden:

```
<label for="firstname">First name</label>
<input name="firstname" id="firstname" type="text"></input>
```

Interagiert der User mit dem Text Field wird ein blinkender Cursor im Inneren platziert und der Rand des Elements färbt sich blau. Ähnlich wie auch Buttons hängt die Darstellung von Input-Fields vom Browser und Betriebssystem ab, das Element lässt sich jedoch beliebig anpassen. So könnte zum Beispiel die Akzentfarbe im Focus State genutzt werden, jedoch ist auch das Standardverhalten benutzbar.

8.0.2.2 Android & iOS

Unter Android sind die input fields abstrakter und besehen nur aus einem Label und einer Grundlinie. Bei Aktivierung floatet das Label nach oben und wird kleiner und der Divider färbt sich. Dieses Verhalten ist im Android SDK eingebunden, es muss lediglich die Farbe angepasst werden. Für die Färbung würde sich auch hier die zuvor gewählte Akzentfarbe anbieten, es muss jedoch getestet werden, ob das Text field mit der gewählten Akzentfarbe funktioniert, wenn der Hintergrund in der Hauptfarbe gefärbt ist.

Unter iOS ist wenig für die visuelle Gestaltung spezifiziert, aber auch hier bestehen die Text Fields aus einem Label und einem Divider. Die Text Fields passen sich visuell nicht an die Akzentfarbe an und können aus Xcode übernommen werden.

8.0.3 Select & Radio Buttons

Select und Radio Buttons unterscheiden sich sowohl in ihrem Aussehen, als auch in ihrer Funktionsweise. Radio Buttons werden genutzt um genau eine aus vielen Optionen auszuwählen, Select oder Checkboxes um beliebig viele aus vielen Optionen zu wählen. Radio Buttons sind auf allen Plattformen rund, Checkboxes eckig und werden mit einem Haken gefüllt, wenn sie aktiviert sind. Auf allen Plattformen gibt es vorgefertigte Lösungen, mit denen gearbeitet werden kann. Gerade im Web gestaltet es sich schwierig, das Standardverhalten zu überschreiben und eine gute Funktionsweise auf allen Endgeräte sicher zu stellen.

8.0.4 Links

Links kommen vor allem im Web vor und verweisen auf eine andere Seite. Sie sollten generell durch eine andere Farbe, zum Beispiel die Akzentfarbe und eine Unterstreichung gekennzeichnet werden:

To maximize the perceived affordance of clickability, color and underline the link text.

Users shouldn't have to guess or scrub the page to find out where they can click.

Da Links in der Regel Textelemente sind, ist eine weitere visuelle Kennzeichnung nicht zwingend nötig und sollte im Rahmen dieses Projektes auch nicht forciert werden.

Kapitel 9

Fazit

9.1 Zusammenfassung

9.2 Fazit

9.3 Schlussbemerkung

Literaturverzeichnis

- [Berman(2010)] Margo Berman. *Street-smart advertising: How to win the battle of the buzz*. Rowman & Littlefield Publishers, 2010.
- [Crozier(1999)] W Ray Crozier. The meanings of colour: Preferences among hues. *Pigment & Resin Technology*, 28(1):6–14, 1999.
- [Cyr et al.(2010)Cyr, Head, and Larios] Dianne Cyr, Milena Head, and Hector Larios. Colour appeal in website design within and across cultures: A multi-method evaluation. *International journal of human-computer studies*, 68(1):1–21, 2010.
- [Eysenck(1941)] Hans J Eysenck. A critical and experimental study of colour preferences. *The American Journal of Psychology*, 54(3):385–394, 1941.
- [Itten(2010)] Johannes Itten. *Kunst der Farbe. Studienausgabe: Subjektives Erleben und objektives Erkennen als Wege zur Kunst*. Christophorus Verlag, 5 edition, 6 2010. ISBN 9783862301614. URL <http://amazon.de/o/ASIN/3862301613/>.
- [Rider(2010)] Rose M Rider. Color psychology and graphic design applications. *Senior Honors Papers*, page 111, 2010.