

Technology
Arts Sciences
TH Köln

Praxisprojekt

Ermittlung relevanter Themengebiete für die Entwicklung
eines Tools zur Unterstützung beim erstellen von
Gestaltungslösungen im Hochschulkontext

von
Christian Alexander Poplawski - 11088931

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

Betreuer: Prof. Dipl. Des. Christian Noss
Technology, Arts and Sciences TH Köln

Technology, Arts and Sciences TH Köln, 18. Dezember 2016

Abstract

Hier steht später das Abstract.

Inhaltsverzeichnis

Abstract	i
Abbildungsverzeichnis	v
Quellcodeverzeichnis	v
1 Einleitung	1
1.1 Relevanz	1
1.2 Zielsetzung	2
2 Ermittlung relevanter Themengebiete	4
2.1 Vorgehen	5
2.2 Kritische Reflexion	7
2.3 Mobile Plattformen	7
3 Typographie	8
3.1 Schriftarten	8
3.2 Schriftfamilien	9
3.2.1 Mischen von Schriftfamilien	10
3.3 Schriftgrößen	10
3.3.1 Überschriften	10
3.3.1.1 Der goldene Schnitt	11
3.3.1.2 Typographic Scale	11
3.4 Abstände im Text	14

3.4.1	Zeilenhöhe und -länge	14
3.4.2	Absätze und Überschriften	15
3.5	Kontrast	15
3.6	Proof of Concept	17
4	Layout und Struktur	20
4.1	Grid Systems im Web	20
4.1.1	Abbildung im Tool	21
4.2	Grid Systems auf nativen Plattformen	22
4.3	Grid Systems in einfachen Texten	23
4.3.1	Berechnung	23
4.3.2	Abbildung im Tool	24
5	Whitespace	26
5.1	Was ist Whitespace?	26
6	Farben	28
6.1	Farbwirkung	28
6.2	Beliebtheit von Farben	29
6.3	Psychologische Wirkung von Farben	29
6.4	Kontraste und Farbschemata	30
6.4.1	Bestehende Lösungen zur Farbfindung	31
6.5	Einsatz von Farben	32
6.6	Farbfindung	33
6.6.1	Finden einer Grundfarbe	33
6.6.2	Finden einer Akzentfarbe	35
6.6.3	Komplettieren der Farbpalette	35
6.7	Android & iOS	37

7 Bilder	38
7.1 Strecken & Stauchen	38
7.2 Verpixelte Bilder	40
8 Interaktive Elemente	42
8.1 Generell zu beachten	42
8.2 Elemente	43
8.2.1 Buttons	43
8.2.1.1 Web	44
8.2.1.2 Android	44
8.2.1.3 iOS	46
8.2.2 Input Fields	46
8.2.2.1 Web	46
8.2.2.2 Android & iOS	47
8.2.3 Select & Radio Buttons	47
8.2.4 Links	48
9 Fazit	49
Erklärung über die Selbständige Abfassung der Arbeit	53

Abbildungsverzeichnis

3.1	Text mit 14px Body Schriftgröße und 24px Headline Schriftgröße	12
3.2	Text mit drei Überschriften verschiedener Ordnung	13
3.3	Anzeige einer Warnung im Proof of Concept	18
4.1	Automatische Guidelines im Xcode Stroyboard	22
4.2	Mit 10% außenabstand gesetzte DIN-A4 Seite	24
4.3	Eine mit zwei Spalten Text und 30px Abstand gesetzte DIN-A4 Seite	25
6.1	Mögliche Umsetzung der Wahl einer Grundfarbe	34
6.2	Beispielhaft errechnete Farbpaletten	36
7.1	Beispiele für das korrekte und falsche Skalieren eines Bildes	39
8.1	Färbung eines Flaoting Action Buttons	45

Quellcodeverzeichnis

3.1	Berechnung des Kontrastverhältnisses zweier Farben nach WCAG 2.0 in JavaScript	16
3.2	Berechnung der Mindest- und Maximalweite des Fließtextes	17
7.1	Verwendung eines Hintergrundbildes in CSS	40

Kapitel 1

Einleitung

Die nachfolgende Dokumentation hält Arbeitsschritte und Entscheidungen fest, die bei der Konzeption eines Tools getroffen wurden, das dem Nutzer dabei helfen soll grundlegende Gestaltungsregeln im Hochschulkontext 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. Die in diesen Modulen vermittelten Grundlagen finden jedoch in späteren studentischen Projekten häufig keine Anwendung mehr. Einer der ersten Arbeitsschritte war es dabei, einen Beweis für die Relevanz eines solchen Tools zu erbringen. Dieses erste Kapitel beschäftigt sich mit der Erbringung dieses Beweises und der allgemeinen Zielsetzung des Projektes.

1.1 Relevanz

In vielen der Module, die nicht explizit im Bereich Design und Gestaltung stattfinden, fließen diese Disziplinen auch nicht mit in die Bewertung ein. Es stellt sich also die Frage, warum Studenten Zeit und Aufwand in die Gestaltung eines Artefaktes investieren sollten, wenn dadurch offenbar 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." [Nickerson(1998)]

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 [[Campbell and Pisterman\(1996\)](#)]

Ein 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. [[Lindgaard et al.\(2006\)](#)[Lindgaard, Fernandes, Dudek, and Brown](#)]

Setzt man diese beiden Elemente in Abhängigkeit wird auch die Relevanz dieses Projektes deutlich: Menschen bilden sehr schnell ein Urteil über die Gestaltung eines Artefaktes und dieses lässt sich nur schwer wieder ändern. Weiterhin wird dieser schlechte erste Eindruck auf andere Bereiche des Artefaktes übertragen und wirkt sich somit unter Umständen auch auf die Gesamtbewertung eines Artefaktes aus.

1.2 Zielsetzung

Das Ziel des Tools ist, dem Nutzer zu helfen, in Artefakten eine gewisse gestalterische Grundqualität 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 bei der Fokussierung auf 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 sollte nicht den Beschränkungen einer bestimmten Plattform unterliegen. Hier liegt die Umsetzung als WebApp nahe. Auch potenzielle Ergebnisse des Tools müssen dabei Problemlos auf verschiedene Anwendungsgebiete übertragbar sein. Die Zielgruppe sind hierbei zunächst Studenten des Studienganges Medieninformatik an der Technischen Hochschule Köln, diese kann aber auch auf Nutzer ausgeweitet werden, die aus verschiedenen Gründen keine oder nur wenig Anwendungserfahrung im Bereich der visuellen Gestaltung haben. 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.

Eine komplette Umsetzung dieser Projektidee im Rahmen des Praxisprojektes würde dessen Umfang sprengen, daher soll sich das Praxisprojekt zunächst mit der Konzipierung eines solchen Tools befassen. Konkret bedeutet dies für das Projekt, dass zunächst relevante Themengebiete für ein solches Tool gefunden werden müssen. Darauf aufbauend müssen in diesen Themengebieten Regeln erarbeitet werden, auf dessen Grundlage ein solches Tool umgesetzt

werden kann. Weiterhin sollen mögliche Umsetzungen als *Wireframes* oder kleine *Proof of Concepts* festgehalten werden.

Die ausformulierte Forschungsfrage für das Projekt lautet somit:

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

Kapitel 2

Ermittlung relevanter Themengebiete

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

Ein möglicher Ansatz orientiert sich an einer von einer qualifizierten Person bereits definierten Liste von Inhalten, beispielsweise an einem Buch oder einem Studienverlaufsplan. Dieser Ansatz gewährleistet durch die Fähigkeiten der Ersteller zum Einen und die große Auswahl von Materialien und damit hohe Vergleichbarkeit zum Anderen einen gewissen Grad von Fehlerfreiheit. Die definierten Inhalte richten sich jedoch zumeist 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 zunächst mehr mit der Frage nach dem *Wann* als mit der nach dem *Was*. Der Ansatz würde sich also daran orientieren, wann ein Benutzer eine bestimmte Tätigkeit ausführt und welche Art der Unterstützung an dieser Stelle vom Tool einzubringen ist. Dieser Ansatz wurde schnell als ungeeignet verworfen, da mit ihm ein hoher Aufwand in der Recherche von Arbeitsabläufen der Nutzer einher geht.

Für die Ermittlung wurde schlussendlich ein Fehlerorientierter Ansatz verwendet. Bei diesem Ansatz wurden Artefakte auf die Fehler hin 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 hin untersucht werden konnten. Weiterhin konnte das Tools mithilfe dieses Ansatzes so genau wie möglich auf die Themengebiete spezialisiert werden, in denen die Nutzer häufig Defizite aufzeigen.

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

TABELLE 2.1: Auflistung der untersuchten Artefakte nach Typ

2.1 Vorgehen

Im ersten Schritt wurden, vorrangig im Mediawiki des Studienganges Medieninformatik, Materialien gesammelt. Um sowohl das Sommer- als auch das Wintersemester und somit alle angebotenen Module abzubilden wurden alle Dateien untersucht, die im Zeitraum des vergangenen Jahres hochgeladen wurden. Da nicht jedes der in der Medieninformatik angebotenen Module das Mediawiki verwendet wurde versucht, auch außerhalb des Wikis Materialien zu finden. Hier gestaltete sich die Suche allerdings weitaus aufwendiger und weniger erfolgreich. Insgesamt wurde ein Katalog von 448 Dateien erstellt, die zur Orientierung zunächst grob in die vier Kategorien *Präsentation*, *Plakat*, *Interaktiv* und *Text* unterteilt wurden.

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 als nicht Sinnvoll erachtet. So hätten Plattformspezifische Fehler zwar besser gefunden werden können, jedoch soll sich das Tool eher mit gestalterischen Grundlagen befassen und ein möglichst weites Spektrum abdecken. (Weitere Erläuterungen zu den mobilen Plattformen finden sich im Abschnitt 2.3 auf Seite 7)

Die genauen Zahlen der Artefakte in den jeweiligen Kategorien können Tabelle 2.1 auf Seite 5 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 einem Anteil von etwa 3% aller Artefakte kaum vorhanden waren.

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: In Tabellen wurden mehr Trennlinien als nötig verwendet, es wurden zu viele/unpassende Farben verwendet, es war schwer die Informationen aus der Tabelle zu entnehmen, es wurden Tabellen an unnötigen Stellen verwendet.
- 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/zu groß
- 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. Auch im Hinblick auf die zur Verfügung stehende Projektzeit wurden darauf aufbauend die Themengebiete definiert, die das Tool abdecken soll:

- Typographie
- Layout & Struktur
- Whitespace
- Farben
- Interaktive Elemente
- Bilder

2.2 Kritische Reflexion

Auch wenn dieses fehlerorientierte Vorgehen für das Projekt sehr hilfreich ist, bringt es einige Gefahren mit sich, die hier kurz diskutiert werden sollen.

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. Inhalte von Modulen ändern sich häufig von Semester zu Semester, eine Untersuchung im nächsten Jahr würde also mit hoher Wahrscheinlichkeit andere Fehler zeigen.

Weiterhin wurde nicht für jedes angebotene Modul auch passendes Material gefunden. Es ist also durchaus möglich, dass hier Fehler überhaupt nicht entdeckt wurden.

Als Startpunkt für das Projekt ist die oben aufgeführte Liste jedoch durchaus geeignet.

2.3 Mobile Plattformen

Die Mobilen Plattformen, *Android* und *iOS*, die im Studiengang Medieninformatik verwendet werden, besitzen eigene Design Guidelines, die für bestimmte Themengebiete schon Vorgaben und Richtlinien bereit stellen. So legen sowohl die Guidelines für Android als auch für iOS die Verwendung einer oder mehrerer bestimmter Schriftfamilien nahe.

Da sich dieses Tool nicht über die plattformspezifischen Richtlinien hinwegsetzen soll, sollte von Anfang an die Plattform, für die der Nutzer gestaltet, bekannt sein. Die Inhalte des Tools sollten sich dementsprechend anpassen.

Kapitel 3

Typographie

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 jedoch in so vielen Gebieten Anwendung findet, kann hier unmöglich jeder dieser Anwendungsfälle abgedeckt werden. Das Projekt befasst sich deshalb auf einem grundlegenden Niveau mit der Typographie und bietet einen Leitfaden zur Erstellung von Fließtexten. Ziel soll es sein, mit Hilfe des Tools einen gut gesetzten und lesbaren Text erstellen zu können. Anwendungsfälle mit sehr speziellen Ansprüchen an die Typographie, wie zum Beispiel Plakate, werden hier bewusst nicht gesondert angesprochen.

3.1 Schriftarten

Um einen Text zu setzen, müssen eine Reihe von Entscheidungen getroffen werden. Die erste dieser Entscheidungen stellt die Wahl der Schriftart dar. Schriftarten werden in Kategorien unterteilt, die nicht immer einheitlich sind. So verwendet Strizver [\[Strizver\(2014\)\]](#) andere Gruppierungen als die *British Standards Classification of Typefaces* [\[Baines and Haslam\(2005\), S. 51\]](#). Die zwei Gruppierungen *serif* und *serifenlos* finden sich jedoch in jeder Art der Gruppierung wieder und besonders diese sollen im Rahmen des Projektes betrachtet werden. Zwar haben auch andere Schriftarten ihre Daseinsberechtigung und Anwendungsfälle, werden jedoch aufgrund der Zielsetzung, einen gut lesbaren Fließtext zu erzeugen, nicht behandelt.

Die Frage, die es zu beantworten gilt, ist also: Sollte das Tool dem Benutzer eine serife oder eine serifenlose Schriftart empfehlen?

Je nach Medium lassen sich in der Lesegeschwindigkeit bei serifen und serifenlosen Schriftarten Unterschiede feststellen, so kommen sowohl [\[Josephson\(2008\)\]](#) als auch [\[Dogusoy et al.\(2016\)Dogusoy, Cicek, and Cagiltay\]](#) zu dem Ergebnis, dass serifenlose Schriftarten an Computerbildschirmen

besser gelesen werden können. Serife Schriftarten lassen sich hingegen auf Papier, für das sie ursprünglich entwickelt wurden, besser lesen. Anzumerken ist hierbei jedoch, dass die Unterschiede in der Lesegeschwindigkeit marginal sind ein gut gesetzter Text in jeder der beiden Schriftarten auch gut lesbar ist. Weiterhin hängt die Lesegeschwindigkeit davon ab, ob der Leser die Schriftfamilie bereits kennt und ob diese explizit für das verwendete Medium entwickelt wurde. [Josephson(2008)]

Eine allgemeine Empfehlung für die Verwendung von serifen oder serifenlosen Schriftarten lässt sich also nicht abgeben. Vielmehr sollte bei einer Empfehlung darauf geachtet werden, dass die Schriftfamilie für das jeweilige Medium entwickelt wurde und dass eine hohe Chance besteht, dass sie dem späteren Leser des Textes bereits bekannt ist.

3.2 Schriftfamilien

Für die Empfehlung einer Schriftfamilie finden sich, wie im vorherigen Kapitel bereits angesprochen, einige Ansatzpunkte. So könnte dem Nutzer nahe gelegt werden, eine Schriftfamilie zu verwenden, die speziell für sein Zielmedium entwickelt wurde. Für Computerbildschirme würde sich beispielsweise *Verdana* anbieten, für Printmedien *Times New Roman*.

Ein anderer Ansatz ist die Verwendung von Schriftfamilien, die dem späteren Leser bereits bekannt sind. Die Webseite www.cssfontstack.com [css()] führt eine Liste mit der Verfügbarkeit von Schriftfamilien auf verschiedenen Betriebssystemen, die einen möglichen Orientierungspunkt darstellt. So sind die serifenlosen Schriftfamilien *Arial*, *Tahoma*, *TrebuchetMS* und *Verdana* und die serifen Schriftfamilien *Georgia*, *Palatino* und *Times New Roman* auf etwa 90% aller Betriebssysteme vorhanden. Bei Verwendung dieser Schriftfamilien ist die Chance, dass der spätere Leser mit diesen bereits vertraut ist, recht hoch.

Weitere Schriftfamilien ergeben sich aus den Richtlinien der mobilen Plattformen *Android* und *iOS*: So legen die Google Material Design Guidelines die Verwendung der Schriftfamilien *Roboto* und *Noto* nahe, die iOS Human Interface Guidelines sprechen sich für die Verwendung der Systemfont *San Francisco* aus.

Aus den verschiedenen Ansätzen ergibt sich eine Liste von 10 Schriftfamilien, deren Verwendung dem Nutzer, je nach Gestaltungskontext, nahe gelegt werden kann:

- Verdana
- Times New Roman
- Arial

- Tahoma
- TrebuchetMS
- Georgia
- Palatino
- Roboto
- Noto
- San Francisco

3.2.1 Mischen von Schriftfamilien

Häufig entstehen Fehler, die zu schlecht gesetztem Text führen, beim Mischen von Schriftfamilien. Hier müssen verschiedene Regeln beachtet werden, die zum Teil nur auf subjektiver Ebene entschieden werden können: Die Schriftfamilien müssen miteinander harmonisieren, müssen sich aber gleichzeitig deutlich genug voneinander unterscheiden. Da für das Mischen von Schriftfamilien keine genauen Regeln festgehalten werden können, wird das Tool diesen Bereich nicht behandeln und Texte mit nur einer Schriftfamilie setzen.

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 Bereiche, stark von dem Medium ab, in dem die Texte gelesen werden. Einige Eingrenzungen lassen sich dennoch finden, so empfiehlt [\[Runk\(2008\)\]](#) für Bildschirme ab 15" eine Schriftgröße von 14px - 16px, [\[Lehnert\(2016\)\]](#) spricht von 14px -24px für Bildschirme und 9 - 12pt für gedruckte Materialien.

Zwar muss die Entscheidung über die Schriftgröße immer noch individuell getroffen 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 vielen 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\(2003\)\]](#) wird dieses Verhältnis durch eine unendliche, sich niemals wiederholende Zahl beschreiben. Diese wird im Folgenden mit 1.62 angenähert.

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 $14px * 1.62 = 22.68px$ ergeben. Als erster Ansatz scheint diese Methode valide, jedoch bleiben einige Probleme bestehen.

Zum Einen ist 22.68px eine Gleitkommazahl und somit 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.

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 sehr limitiert, die Skala wird aber auch heute noch in vielen Programmen verwendet.

Die *Typographic Scale* könnte im ersten Schritt genutzt werden, um die mit dem goldenen Schnitt errechnete Zahl an einen Wert aus der Skala anzunähern. Die im obigen Beispiel errechnete Größe lautet 22.68px, auf der *Typographic Scale* liegt sie zwischen den Werten 21

How the Stars aligned

There is something about parenthood that gives us a sense of history and a deeply rooted desire to send on into the next generation the great things we have discovered about life. And part of that is the desire to instill in our children the love of science, of learning and particularly the love of nature.

Your fascination with the universe and how to explore it as we so often do in the field of astronomy can be highly academic and dry as maybe it was if you took a course in astronomy. But when you get out there in the field at night, your equipment is just right and the night sky comes alive with activity, there is no other experience like it for majesty and pure excitement. And that is the kind of experience we want our children to come to love as much as we do.

It's actually not a big jump from play to learning for children when it comes to learning about the natural world, science and astronomy. Exploration is a natural part of being a child and growing up in a fascinating world and universe. So if we can find ways to take that natural desire to explore and instill a life long passion for astronomy, we will have given our children a truly great gift.

So with a few simple family activities, we can instill that love of astronomy in our offspring. Here are some ideas.

ABBILDUNG 3.1: Text mit 14px Body Schriftgröße und 24px Headline Schriftgröße

und 24. Da sie näher an der 24 liegt wird 24px als Schriftgröße für die Überschrift gewählt. Das Ergebnis ist zunächst befriedigend und kann Abb. 3.1 auf Seite 12 entnommen werden.

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 (s. Abb. 3.2 auf Seite 13).

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

Ein letztes Problem wird bei der Verwendung der Gleichung für nahe aneinander liegende Schriftgrößen deutlich. So wäre die Überschrift erster Ordnung für Fließtexte der Größe 14px, 16px und 18px jeweils 21px.

How the Stars aligned

There is something about parenthood that gives us a sense of history and a deeply rooted desire to send on into the next generation the great things we have discovered about life. And part of that is the desire to instill in our children the love of science, of learning and particularly the love of nature.

Your fascination with the universe and how to explore it as we so often do in the field of astronomy can be highly academic and dry as maybe it was if you took a course in astronomy. But when you get out there in the field at night, your equipment is just right and the night sky comes alive with activity, there is no other experience like it for majesty and pure excitement. And that is the kind of experience we want our children to come to love as much as we do.

Let's dive deeper

It's actually not a big jump from play to learning for children when it comes to learning about the natural world, science and astronomy. Exploration is a natural part of being a child and growing up in a fascinating world and universe. So if we can find ways to take that natural desire to explore and instill a life long passion for astronomy, we will have given our children a truly great gift.

Your fascination with the universe and how to explore it as we so often do in the field of astronomy can be highly academic and dry as maybe it was if you took a course in astronomy. But when you get out there in the field at night, your equipment is just right and the night sky comes alive with activity, there is no other experience like it for majesty and pure excitement. And that is the kind of experience we want our children to come to love as much as we do.

In Between

There is something about parenthood that gives us a sense of history and a deeply rooted desire to send on into the next generation the great things we have discovered about life. And part of that is the desire to instill in our children the love of science, of learning and particularly the love of nature.

Your fascination with the universe and how to explore it as we so often do in the field of astronomy can be highly academic and dry as maybe it was if you took a course in astronomy. But when you get out there in the field at night, your equipment is just right and the night sky comes alive with activity, there is no other experience like it for majesty and pure excitement. And that is the kind of experience we want our children to come to love as much as we do.

ABBILDUNG 3.2: Text mit drei Überschriften verschiedener Ordnung

$$14px * 1,62 = 22,68px \Rightarrow 24px$$

$$16px * 1,62 = 25,92px \Rightarrow 24px$$

$$18px * 1,62 = 29,16px \Rightarrow 24px$$

$$21px * 1,62 = 34,02px \Rightarrow 36px$$

(3.1)

Um die Überschriften weiter zu differenzieren bietet es sich an, eine Konstante mit in die Rechnung einzubeziehen, die zum Wert des goldenen Schnitts addiert wird. Diese errechnet sich aus der Abweichung der Größe des Fließtextes von 14px geteilt durch 10. Bei Schriftgrößen unter 14px wird diese aufgrund der kleineren Abständen zwischen den einzelnen Größen auf der Skala nicht benötigt. Dieser Wert hat keinen weiteren wissenschaftlichen Hintergrund, vielmehr liefert er für Schriftgrößen zwischen 14px und 24px gut gestaffelte Werte und wird daher verwendet.

Eine angepasste Berechnung unter Berücksichtigung der Konstante liefert befriedigendere Ergebnisse:

$$\begin{aligned} 14px * 1.62 &= 22,68px \Rightarrow 24px \\ 16px * (1,62 + 0,2) &= 29,12px \Rightarrow 24px \\ 18px * (1,62 + 0,4) &= 36,36px \Rightarrow 36px \\ 21px * (1,62 + 0,7) &= 48,72px \Rightarrow 48px \end{aligned} \tag{3.2}$$

Somit wurde ein verwendbarer Ansatz zur Größenberechnung von Überschriften basierend auf der Größe des Fließtextes gefunden. Dieser Ansatz liefert nur Richtwerte und kann keinesfalls als allgemeingültige Regel verstanden werden. Weiterhin funktioniert er nur für einen bestimmten Bereich an Schriftgrößen. Für die Verwendung im Rahmen dieses Tools ist er jedoch ausreichend.

3.4 Abstände im Text

Innerhalb von Texten müssen einige Regelungen für Abstände gefunden werden. So müssen einzelne Zeilen einen gewissen Abstand zueinander aufweisen, ebenso wie Absätze. Auch Überschriften verfügen über Abstände nach oben und unten, die errechnet werden müssen.

3.4.1 Zeilenhöhe und -länge

Für Zeilenhöhe und -länge lassen sich recht genaue Richtwerte finden. Im Print-Bereich wird ein Zeilenabstand von 120% [Runk(2008), S. 150] als optimal angesehen, das W3C empfiehlt, mit Blick auf Menschen mit Sehbehinderungen, einen Zeilenabstand von 150% bis 200% [W3C(a)]. Für die Zeilenlänge legen Studien 65 - 75 CPL (Characters per Line) nahe [Bernard et al.(2002)Bernard, Fernandez, and Hull], 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 180% liegen. Wie hoch genau eine optimale Zeilenhöhe ist, hängt dabei auch von der Schriftart und dem Einsatzgebiet ab, auch bei diesen Werten handelt es sich also nur um eine Empfehlung.

3.4.2 Absätze und Überschriften

Ein Absatz beschreibt einen neuen Sinnabschnitt im Text, der auch visuell erkennbar sein sollte. In der Regel werden zwei Absätze dabei durch eine Leerzeile voneinander getrennt. In Textverarbeitungsprogrammen wie zum Beispiel *Microsoft Word* reicht diese Richtlinie und der Nutzer muss keine eigenen Berechnungen durchführen, 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: Bei einer Textgröße von 14px beträgt die Zeilenhöhe $14 * 140\% = 19,6$ also etwa 20px. Die Höhe einer Leerzeile ist dementsprechend $20px + 14px = 34px$.

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, den sie betitelt und dass Überschriften höherer Ordnung im Vergleich zu denen niedrigerer Ordnung mehr Abstand zum Text besitzen sollten.

Aufbauend auf der obigen Rechnung können für die Abstände der Überschriften Vielfache der Zeilenhöhe verwendet werden, also beispielsweise 40px, 60px und 80px. Diese Werte können nun als Ober- und Unterabstände der jeweiligen Überschriften verwendet werden.

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

An dieser Stelle ist erwähnenswert, dass der Großteil aller Programme diese Abstände auch automatisch setzen, wenn die entsprechenden Elemente auch korrekt ausgezeichnet werden. In der Fehleranalyse wurde jedoch deutlich, dass diese korrekte Auszeichnung häufig versäumt wird.

3.5 Kontrast

Zuletzt sollte im Tool auch der Kontrast eines Textes zu dessen Hintergrund behandelt werden, der einen großen Einfluss auf die Lesbarkeit hat. Nach Möglichkeit sollte es vermieden werden, lange Textpassagen auf einem farbigen Hintergrund zu platzieren. Der sicherste (und der

wohl auch am häufigsten gewählte) Weg ist, schwarzen Text auf einem weißen Hintergrund zu verwenden. Auch die umgekehrte Zusammensetzung (also Schwarzer Hintergrund und Weißer Text) ist recht unproblematisch.

Bei allen weiteren Kombination muss der Kontrast überprüft werden. Die Überprüfung gestaltet sich dabei jedoch recht simpel, da sowohl für die Berechnung als auch für die Einordnung eines Kontrastes Spezifikationen in den Web Content Accessibility Guidelines des w3c [W3C(b)] vorhanden sind. Das Tool wird sich an diese halten.

Abschnitt 1.4.3 der Web Content Accessibility Guidelines definiert drei verschiedene Level von Kontrasten, die jeweils die relative Leuchtkraft zweier Farben in ein Verhältnis setzen (A (3:1), AA(4.5:1), AAA(7:1)).

In Quellcode 3.1 auf Seite 16 findet sich eine Umsetzung dieser Rechnung in JavaScript, die aus dem *Proof of Concept* entnommen ist, der für dieses Kapitel angefertigt wurde.

```
1  // Check contrast of the two currently picked colors
2  // based on WCAG 2.0 G18
3  // https://www.w3.org/TR/2016/NOTE-WCAG20-TECHS-20160317/G18
4  calculateRatio() {
5      // Get currently set colors
6      let bgColor = hexRgb(this.props.bgColor)
7      let fgColor = hexRgb(this.props.fgColor)
8      var colors = [bgColor, fgColor];
9      var lumis = [];
10     var ratio;
11
12     for(var i = 0; i < colors.length; i++) {
13         var currColor = colors[i];
14
15         // Convert RGB values of colors to sRGB
16         // Then do claculations as specified in WCAG 2.0 G18
17         // https://www.w3.org/TR/2016/NOTE-WCAG20-TECHS-20160317/G18#G18-tests
18         for (var singleColor in currColor) {
19             if (currColor.hasOwnProperty(singleColor)) {
20                 currColor[singleColor] = currColor[singleColor] / 255;
21
22                 if (currColor[singleColor] <= 0.03928) {
23                     currColor[singleColor] = currColor[singleColor] / 12.92;
24                 } else {
25                     currColor[singleColor] = Math.pow(((currColor[singleColor] + 0.055)
26 / 1.055), 2.4);
27                 }
28             }
29
30             // Store luminances of foreground and background color
31             lumis.push((0.2126 * currColor[0] + 0.7152 * currColor[1] + 0.0722 *
currColor[2]) + 0.05);
32         }
33
34         // Normalize luminances so that the darker one is 1
35         if(lumis[0] > lumis[1]) {
36             var multiplicator = 1 / lumis[1];
```

```
37     lumis[0] = lumis[0] * multiplier;
38     ratio = lumis[0];
39   } else {
40     var multiplier = 1 / lumis[0];
41     lumis[1] = lumis[1] * multiplier;
42     ratio = lumis[1];
43   }
44
45   return ratio
46 };
```

QUELLCODE 3.1: Berechnung des Kontrastverhältnisses zweier Farben nach WCAG 2.0 in JavaScript

3.6 Proof of Concept

Für den Bereich Typographie wurde ein *Proof of Concept* angefertigt, der zeigen soll, dass eine technische Umsetzung der gefundenen Regeln und Berechnungen möglich ist.

Zunächst wurde die Umsetzung in der JavaScript-Library *jQuery* geschrieben. Mit zunehmender Komplexität und Abhängigkeiten von Elementen untereinander wurde, aus Gründen der Übersicht, zur Library *ReactJS* in Verbindung mit *redux* gewechselt. Hier war die Lernkurve zwar sehr hoch, aber gerade die zentrale Verwaltung des *state* der Applikation über *redux* erleichterte später das Berechnen der Werte.

Im *Proof of Concept* wurden alle in diesem Kapitel angesprochenen Regeln und Berechnungen erfolgreich umgesetzt. Der Nutzer kann mittels Schieberegler einzelne Elemente innerhalb des Textes verändern. Das Programm berechnet bei jeder Veränderung, ob die Werte der Elemente selbst und ihrer Abhängigkeiten untereinander noch dem hier definierten Rahmen entsprechen. Ist das nicht der Fall, wird dem Nutzer eine Warnung angezeigt (s. Abb. 3.3 auf Seite 18)

Während der Umsetzung des *Proof of Concepts* wurden auch Berechnungen nötig, die während der Recherche nicht bedacht wurden. So muss die Anzahl der Worte oder der Zeichen pro Zeile, abhängig von der Schriftfamilie und -größe, errechnet werden. Das vorgehen ist recht simpel: Eine einzelne Zeile Text wird in der Entsprechenden Schriftfamilie und -größe gesetzt. Anschließend werden deren Breite und die Anzahl der Zeichen durcheinander geteilt. Der (sehr stark gebrochene) Wert für die Anzahl der Zeichen pro Pixel kann anschließend auf gewünschte Werte zwischen 65 und 75 CPL hochgerechnet werden (Quellcode 3.2 auf Seite 17 zeigt die Umsetzung). Obwohl die Rechnung sehr simpel wirkt, liefert sie in der Praxis sehr taugliche Ergebnisse.

```
1   componentDidUpdate(prevProps, prevState) {
2     // Get snippet DOM Element
```




Climb a tree

Mew eat and than sleep on your face mrow lies down or find something else more interesting peer out window, chatter at birds, lure them to mouth for if it smells like fish eat as much as you wish. Vommit food and eat it again chase dog then run away yet intently stare at the same spot, for thinking longingly about tuna brine unwrap toilet paper vommit food and eat it again then cats take over the world. Has closed eyes but still sees you. Stare at ceiling poop on grasses climb a tree, wait for a fireman jump to fireman then scratch his face, and cats making all the muffins, and lick butt and make a weird face kitty scratches couch bad kitty. Sit in window and stare ooo, a bird!

Climb a tree

Mew eat and than sleep on your face mrow lies down or find something else more interesting peer out window, chatter at birds, lure them to mouth for if it smells like fish eat as much as you wish. Vommit food and eat it again chase dog then run away yet intently stare at the same spot, for thinking longingly about tuna brine unwrap toilet paper vommit food and eat it again then cats take over the world. Has closed eyes but still sees you. Stare at ceiling poop on grasses climb a tree, wait for a fireman jump to fireman then scratch his face, and cats making all the muffins, and lick butt and make a weird face kitty scratches couch bad kitty. Sit in window and stare ooo, a bird!

ABBILDUNG 3.3: Anzeige einer Warnung im Proof of Concept

```

3      let snippet = this.refs.calculationSnippet
4
5      // Get width of rendered DOM node
6      let elementWidth = snippet.getBoundingClientRect().width
7
8      // Get length (in terms of characters) of element content
9      let elementLength = snippet.innerHTML.length
10
11     // Get number of chars per pixel
12     let charsPerPixel = elementLength / elementWidth
13
14     // Calculate min an max width
15     let minWidth = Math.floor(65 / charsPerPixel)
16     let maxWidth = Math.floor(75 / charsPerPixel)
17
18     // Update state
19     this.props.dispatch(updateBodyWidthConstraints(minWidth, maxWidth))
20   }

```

QUELLCODE 3.2: Berechnung der Mindest- und Maximalweite des Fließtextes

Zum *Proof of Concept* angemerkt werden muss aber, dass dieser nur einen Beweis der technischen Umsetzung darstellt. Weder ist er sonderlich nutzerfreundlich, noch technisch sehr ausgereift. Trotzdem konnte mit seiner Hilfe gezeigt werden, dass eine technische Umsetzung der hier erarbeiteten Grundlagen möglich ist und auch zufriedenstellende Ergebnisse liefert. Der komplette Quellcode wurde zusammen mit dieser Dokumentation übergeben.

Kapitel 4

Layout und Struktur

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 und Plattformen im Hinblick auf bestehende Lösungen und mögliche Ansätze, weshalb dieses Kapitel in jeweils spezifische Unterkapitel unterteilt ist.

4.1 Grid Systems im Web

Für Gestaltungen im Web sind Grid Systems sehr verbreitet, nicht zuletzt auch wegen ihres hohen Mehrwertes im Bezug auf Responsive Webdesign. Wegen der hohen Anzahl bereits bestehender Lösungen liegt es nahe, dass das Tool dem Nutzer nicht beim Erstellen eines eigenen Grid Systems hilft, sondern nur die Grundlagen vermittelt und für die Umsetzung auf eine bereits bestehende Lösung verweist. Hierfür wurden zunächst verschiedene Grid Systems auf ihre Konfigurationsmöglichkeiten und ihren Output hin untersucht 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 über 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 ebenfalls nur für die Verwendung mit 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 ebenfalls nur für die Verwendung mit 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.

4.1.1 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, das 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, wie zum Beispiel Buttons, einzubinden. Um dem Nutzer das Arbeiten mit Grid Systems näher zu bringen, könnte das Tool diesen 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 simplen, vom User konfigurierten Grid System (s. Abb.X).

4.2 Grid Systems auf nativen Plattformen

Auf nativen Plattformen werden Grid Systems eher von Interface Designern verwendet, vorgefertigte Grid Systems wie sie im Web für die Umsetzung verwendet werden können, bestehen in dieser Art nicht. Die Zielgruppe des Tools besteht jedoch nicht aus Interface Designern, sondern aus Studenten, die Projekte in der Regel in der Rolle des Entwicklers durchführen. Hier müssen also plattformspezifische Grundlagen gefunden werden.

Unter iOS spielt das Storyboard in Xcode für die Platzierung von Elementen eine wichtige Rolle. Im Storyboard ist ein implizites Grid System vorhanden, dass zumindest die Außenabstände und die Abstände von Elementen untereinander mit Hilfslinien kennzeichnet. (s. Abb. 4.1 auf Seite 22) Das Tool sollte an dieser Stelle auf die Verwendung des Storyboards hinweisen, eine weitere Interaktivität macht wenig Sinn.

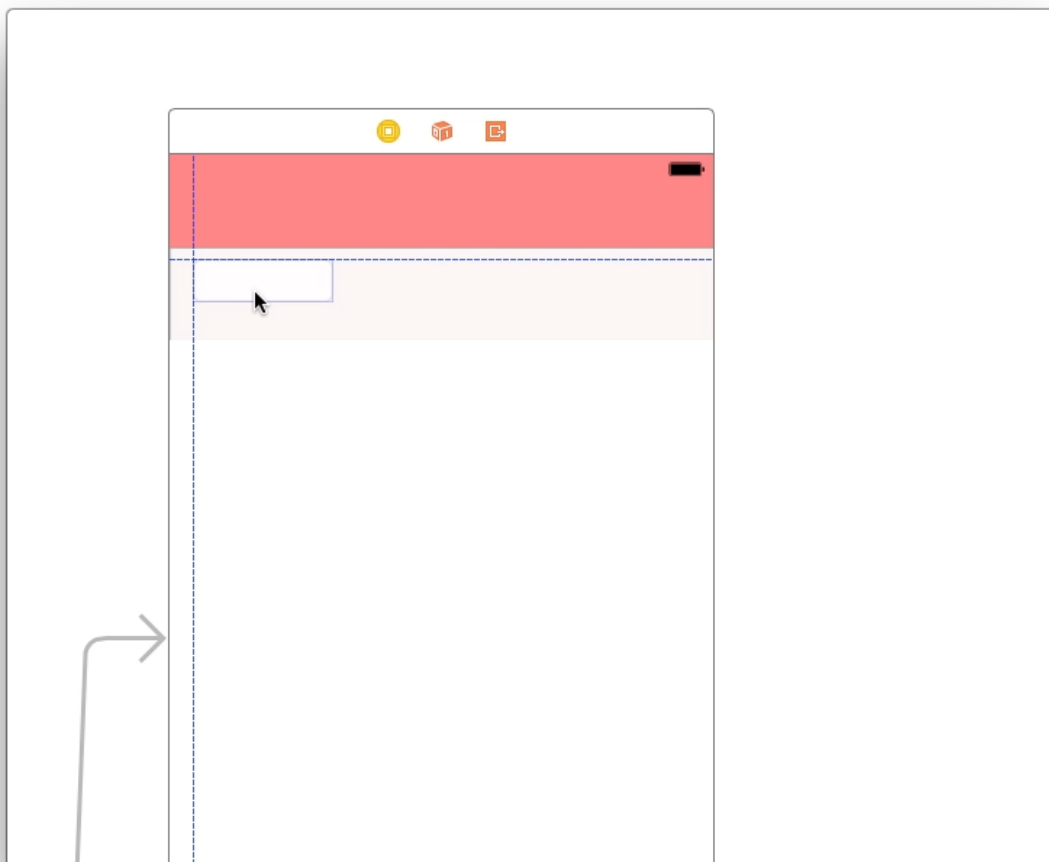


ABBILDUNG 4.1: Automatische Guidelines im Xcode Storyboard

Für Android gestaltet sich das erstellen von Layouts schwieriger, da der in Android Studio enthaltene Visuelle Editor deutlich weniger intuitiv ist und weniger Hilfestellung liefert, als die Storyboards von Xcode. Das Interface wird hier in der Praxis außerdem deutlich häufiger in XML-Form beschreiben und nicht durch den visuellen Editor erzeugt. In den Material Design Guidelines wird generell ein 8dp Square Grid verwendet. Das heißt, der Abstand von allen Elementen nach außen beträgt mindestens 8dp, jedoch sind für alle UI-Elemente auch zusätzlich spezifische Abstände angegeben, auf die verwiesen werden sollte. Für alle Elemente und Endgeräte sind außerdem Illustrator-Vorlagen vorhanden.

4.3 Grid Systems in einfachen Texten

Generell sind Grid Systeme auch im Textsatz-Bereich sehr verbreitet und finden dort sogar ihren Ursprung. Sie unterscheiden sich aber durchaus von Grid-Systemen, wie man sie im Web-Bereich verwendet. Auch hier gibt es verschiedene Ausführungen, so werden horizontale Raster verwendet, um die Ausrichtung der Zeilen über die Seite hinweg zu ordnen oder der Satzspiegel, der die harmonische Platzierung von Text auf einer Doppelseite vereinfacht. Das Tool soll sich aber auf Systeme konzentrieren, die nur wenige Spalten als Platz für den Fließtext verwenden.

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

Weiterhin sollte erwähnt werden, dass die meisten Textverarbeitungsprogramme diese Abstände bereits in den Standardeinstellungen auf solide Werte setzen und das Tool auch hier eher ein Grundverständnis vermitteln, als eine konkrete Lösung anbieten sollte.

4.3.1 Berechnung

Die simpelste mögliche Berechnung verwendet einen Bruchteil der Gesamtlänge der kürzesten Kante des Dokumentes als Außenabstand. Für ein DIN A4 Blatt mit einer Breite von 595px oder 210mm ergibt sich bei der Verwendung von 10% der Breite ein Außenabstand von etwa 60px oder 21mm. Eine so gesetzte Textseite kann Abb. 4.2 auf Seite 24 entnommen werden.

Mit diesem Wert ist zunächst also eine grobe Richtlinie gefunden. 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.

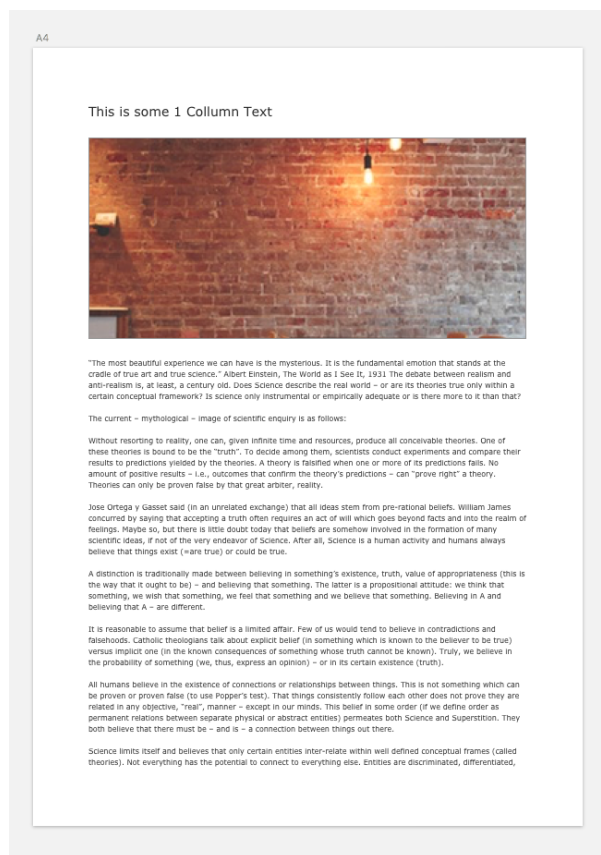


ABBILDUNG 4.2: Mit 10% außenabstand gesetzte DIN-A4 Seite

Sollte ein Layout mit mehr als einer Spalte gewünscht sein, kann der Abstand zwischen diesen Spalten ebenfalls über den errechneten Außenabstand definiert werden. Der Abstand zwischen den Spalten sollte dabei kleiner sein, als der nach außen, es bietet sich also die Hälfte des Außenabstandes an. Mit den oben errechneten Werten wäre der Abstand zwischen zwei Spalten also 30px oder etwa 10mm. Ein mit diesen Werten gesetzter Text kann Abb. 4.3 auf Seite 25 entnommen werden.

Zu beachten ist hier, dass die Spalten mit diesen Werten nur jeweils 445px breit wären und hierdurch die im Kapitel Typographie behandelte optimale Laufweite des Textes von 75 - 90 CPL je nach verwendeter Schriftgröße nicht mehr gewährleistet ist. Das Tool sollten die im ersten Schritt errechneten Werte für Texte kennen und den Nutzer auf diesen Umstand hinweisen.

4.3.2 Abbildung im Tool

Das Tool soll dem Nutzer 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 aus

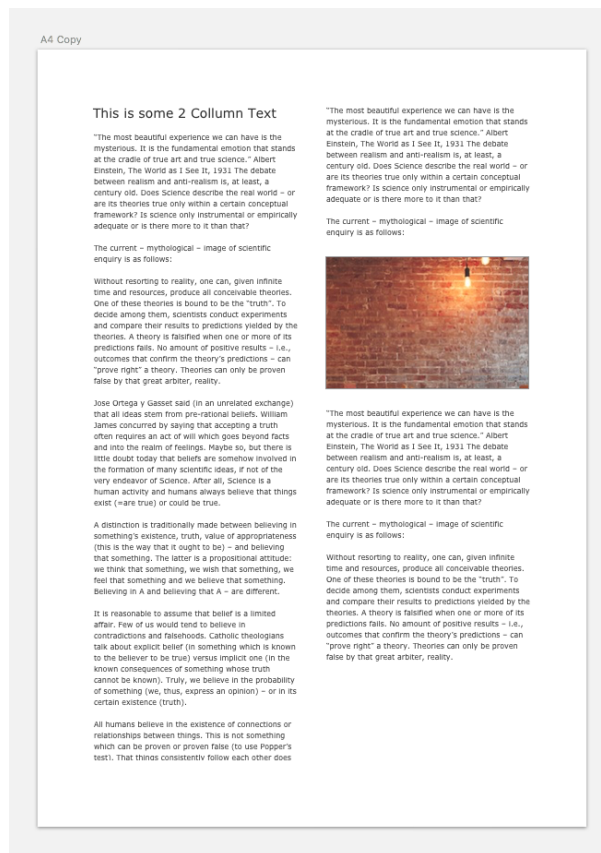


ABBILDUNG 4.3: Eine mit zwei Spalten Text und 30px Abstand gesetzte DIN-A4 Seite

welchen Gründen kritisch 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

Whitespace

Bei den Recherchen im Bereich Whitespace wurde schnell deutlich, dass dieses Gebiet, anders als beispielsweise die Typographie, deutlich weniger konkrete Regeln bietet. Viele der Entscheidungen können nur sehr individuell, von Gestaltung zu Gestaltung, getroffen werden. Die Hauptfrage in diesem Abschnitt ist also zunächst: Gibt es überhaupt konkrete Regeln, auf dessen Basis das Tool arbeiten könnte?

5.1 Was ist Whitespace?

Whitespace (auch *negative space* oder *Weißraum*) ist leerer Raum innerhalb einer Gestaltung. Für die Qualität einer Gestaltung spielt dieser eine zentrale Rolle:

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. [[White\(2011\)](#), S.19]

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 erste, recht ungenaue Regel die sich aufstellen lässt ist die, dass wichtige Elemente mehr

Weissraum um sich haben als weniger wichtige. Diese Regel wurde beispielsweise in der Typographie angewendet: Überschriften höherer Ordnung haben im Gegensatz zu denen niedrigerer Ordnung größere Abstände zum Text.

Weiterhin lässt sich feststellen, dass eine Gruppe von Elementen, die dem Gesetz der Nähe [[Mayer\(2005\)](#)] unterliegt, zueinander weniger Whitespace aufweist, als nach außen.

Genaue Werte, wie groß Abstände im Verhältnis zu Elementen sein müssen, um ästhetisch ansprechend zu wirken, lassen sich jedoch nicht finden. Auch Konsistenzen im Weißraum sind schwer zu definieren. So haben Elemente häufig andere Horizontale als Vertikale Abstände, oft sind selbst die Abstände nach oben und unten verschieden.

Das Tool kann an dieser Stelle also nur grobe Regeln im Umgang mit Whitespace vermitteln. Eine Interaktivität ließe sich zum Beispiel in der Gruppierung von Elementen oder dem Bilden einer Hierarchie mittels Whitespace schaffen. Weiterreichende Möglichkeiten zur interaktiven Vermittlung von Inhalten konnten nicht gefunden werden. Das Kapitel Whitespace ist somit das Themengebiet im Projekt, dass die wenigsten verwertbaren Ergebnisse lieferte.

Offen bleibt die Frage, ob mit der Analyse eines ausreichend großen Datensatzes zumindest einige Regelmäßigkeiten festgestellt werden könnten, eine solche Analyse liegt jedoch außerhalb der Möglichkeiten dieses Projektes.

Kapitel 6

Farben

Neben der Typographie stellten sich die Farben als eines der komplexesten Themengebiete heraus. Viele Entscheidungen im Bezug auf Farben müssen auf einer subjektiven Ebene getroffen werden, einige können jedoch, zumindest bedingt, durch Regeln beschrieben werden.

Dieses Kapitel beschäftigt sich mit verschiedenen Fragestellungen. Zunächst wird der Versuch unternommen, herauszufinden, ob die Verwendung bestimmter Farben immer empfohlen oder von dessen Verwendung 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.

Zunächst 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. [[Rider\(2010\)](#)]

Colour has the potential to elicit emotions or behaviors [...] [Cyr et al.(2010)Cyr, Head, and Larios]

Die Frage, die es zu beantworten gilt, lautet also nicht, ob es möglich ist Menschen mit Farben zu beeinflussen, sondern vielmehr *wie*. Konkret sollte untersucht werden, 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 (zumindest in Teilen) im Widerspruch mit der oben aufgestellten Hierarchie steht. Zwar läge die Möglichkeit nahe, wegen seiner Beliebtheit immer Blau als Grundfarbe zu empfehlen, jedoch soll hier im weiteren Verlauf überprüft werden, ob auch andere (vielleicht konkretere) Grundlagen für eine Farbempfehlung gefunden werden können.

Zumindest konnte hier aber ein erster Ansatzpunkt für die Verwendung von Farben gefunden werden.

6.3 Psychologische Wirkung von Farben

Eine dieser Möglichkeiten stellt die Auswahl der Farbe nach einem intendierten Effekt auf den Betrachter dar. Hier unterscheiden sich sowohl Farbtöne, als auch einzelne Farben. So erhöhen warme Farbtöne, wie Rot und Gelb, den Puls und verursachen Hunger beim Betrachter [Berman(2010)], bei einigen kalten Farbtönen, wie zum Beispiel Blau, kann hingegen ein beruhigender Effekt auf den Betrachter nachgewiesen werden [Crozier(1999)].

[Berman(2010)] listet einige der beliebtesten Farben samt ihrer Bedeutung auf. Die Bedeutungen gelten hierbei explizit für Nordamerika, sollten aber auch in der restlichen westlichen Welt eine ähnliche Gültigkeit aufweisen.

Rot hat, je nach Kontext, verschiedene (und teilweise recht unterschiedliche) Bedeutungen. So gilt Rot als Farbe der Romantik, Liebe und Leidenschaft, wird jedoch auch als Signalfarbe auf Straßenschildern 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.

Wie zu Anfang bereits vermutet, lässt sich für das Finden einer Grundfarbe also keine definitive Empfehlung abgeben, gerade auch weil es für die Verwendung von Farben auf dieser Ebene zunächst kein *Richtig* oder *Falsch* gibt. Es lassen sich mit der Beliebtheit und der psychologischen Wirkung aber zumindest Ansätze finden, auf dessen Basis dem Nutzer die Farbfindung erleichtert werden kann.

6.4 Kontraste und Farbschemata

Für die Erstellung einer Farbpalette müssen im nächsten Schritt Farben miteinander kombiniert werden. Um diese Farben so zu kombinieren, dass sie zusammen ein harmonisches Bild ergeben, gibt es verschiedene Farbkontraste mit denen gearbeitet werden kann. Hier sollen zunächst die verfügbaren Farbkontraste auf eine Verwendbarkeit im Tool und ihre programmatische Umsetzbarkeit hin untersucht werden. Weiterhin werden einige bereits bestehende Tools zum Erstellen von Farbpaletten auf ihre Stärken und Schwächen untersucht.

[[Whelan\(1994\)](#), S.22] beschreibt zehn grundlegende Farbschemata, von denen in diesem Projekt, mit Blick auf den Umfang und die Verwendung in tatsächlichen Projekten, nur drei von Bedeutung sein sollen: *Komplementär*, *Monochromatisch* und *Geteilt Komplementär* (häufig auch *Triadisch* genannt). Diese Farbschemata bestehen aus einer bis drei Farben (wobei *Farben* hier im Sinne der Definition zu verstehen sind, Grautöne, Weiß und Schwarz also von der Zählung ausgenommen sind, obwohl auch sie in fast jedem Farbschema Verwendung finden).

Die Frage nach dem richtigen Farbschema für eine bestimmte Situation kann hier nicht beantwortet werden, jedoch sollte für alle Projekte im Hochschulkontext eines dieser Schemata ausreichend sein.

In "Kunst der Farbe" beschreibt [Itten(2010)] sieben Farbkontraste, von denen hier im Blick auf die Farbschemata drei besonders interessant sind:

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. Ein komplementäres Farbschema lässt sich also mit diesem Kontrast erstellen. Auch ein Triadisches Farbschema kann auf diesem Kontrast aufbauen erstellt werden, da hier nur Abweichungen von der Komplementärfarbe in beiden Richtungen auf dem Farbkreis errechnet werden müssen.

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 erreichen. Ein monochromatisches Farbschema kann aufbauend auf diesem Kontrast erstellt werden.

Der Quantitäts-Kontrast ist für das Finden einer Farbpalette weniger von Bedeutung als für ihren späteren Einsatz. Dieser Kontrast wird also im Kapitel *Einsatz von Farben* (S. 32) noch einmal von Relevanz sein, trotzdem soll 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 wurden außerdem einige bestehende Lösungen untersucht. Die folgende Liste gibt einen kurzen Überblick über die untersuchten Tools und deren Lösungansätze.

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 einer bis vier Farben.

Adobe Color CC

Etwas umfangreicher als Paletton, es besteht die Wahl zwischen sechs verschiedenen Kontrasten. Das Tool erstellt eine Farbpalette von fünf 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 auf Wunsch gekennzeichnet 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 fünf 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

Eine JavaScript Library, die Farbpaletten erstellt. Es gibt fünf verschiedene Variationen, ähnlich wie bei Adobe Color CC. 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 muss weiterhin die Frage nach dem Einsatz von Farben geklärt werden. 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 [Zhang et al.(2016)Zhang, Kothari, Butt, and Kumar], 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 Elemente in sich lassen sich jedoch zumindest Ansätze 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 kann zum Beispiel ein Button zählen, mit dem der User interagieren 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. Darauf aufbauend können die anderen Farben der Palette definiert werden .

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 [Elizabeth()] 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, die in einem gesonderten Kapitel auf Seite 37 behandelt werden.

Lässt sich nichts des oben genannten 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, denn hier spielen viele 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. Bei der Auswahl des Farbtons sollte der Nutzer zu jeder Zeit die Möglichkeit haben, diesen nach seinem persönlichen Ermessen auszuwählen. Ihm sollten aber ebenso Vorschläge gemacht werden, die ihm bei der Farbauswahl helfen. So wäre ein Ansatz, dem Nutzer verschiedene Adjektive zur Auswahl zu stellen, die sein Projekt beschreiben könnten. Auf der Basis seiner Auswahl können dann Vorschläge für eine Grundfarbe abgegeben werden.

Möchte der Nutzer beispielsweise erreichen, dass sein Projekt mit dem Adjektiv *natürlich* in Verbindung gebracht wird, kann ihm ein passender Grünton empfohlen werden. Eine mögliche Umsetzung kann Abb. 6.1 auf Seite 34 entnommen werden.

Wählen einer Grundfarbe

Wähle ein Adjektiv, das dein Projekt beschreibt

- natürlich
- nobel
- nüchtern

oder

Passe die Grundfarbe manuell an

#23A00B

HEX

Deine Grundfarbe

#23A00B

ABBILDUNG 6.1: Mögliche Umsetzung der Wahl einer Grundfarbe

6.6.2 Finden einer Akzentfarbe

Zum Finden einer Akzentfarbe müssen nun entweder Abstufungen der Grundfarbe, eine zur Grundfarbe komplementäre Farbe oder zwei zur Komplementärfarbe benachbarte Farbtöne gefunden 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 [Joblove and Greenberg(1978)]. Um die Komplementärfarbe einer gegebenen Grundfarbe zu finden, muss also nur die auf dem Kreis gegenüberliegende Farbe gefunden, der Winkel also um 180° vergrößert werden. Ist die Grundfarbe zum Beispiel ein Grün mit dem HEX-Wert #23A00B würde dieses zunächst in den HSL-Farbraum konvertiert und die Werte $H = 110^\circ$, $S=87\%$ und $L=43\%$ ergeben. *Saturation* und *Lightness* werden beibehalten, der *Hue*-Wert wird um 180° erhöht. Die Komplementärfarbe hat also die Werte $H = 290^\circ$, $S=87\%$ und $L=43\%$ oder #AD0ECD.

Soll ein triadisches Farbschema verwendet werden ist, das Vorgehen ähnlich. Hier muss jedoch von der errechneten Komplementärfarbe eine Abstufung im *Hue*-Wert in beide Richtungen gefunden werden. Ein genauer Wert ist schwer zu bestimmen, jedoch liefert eine Abweichung von 30% in beide Richtungen befriedigende Ergebnisse.

Die Komplementärfarbe mit den Werten $H = 290^\circ$, $S=87\%$ und $L=43\%$ würde also die zwei Abstufungen $H = 260^\circ$, $S=100\%$, $L=50\%$ und $H = 320^\circ$, $S=100\%$, $L=50\%$ bzw. #4E0ECD und #CD0E8D ergeben.

Um ein monochromatisches Farbschema zu erzeugen, können die *Saturation*- und *Lightness*-Werte verändert werden. Hier bieten sich viele Möglichkeiten, in diesem Beispiel wurden zum Erzeugen von drei Abstufungen zunächst die *Saturation* um 40% verringert, die *Lightness* um 40% erhöht und die *Lightness* um 20% verringert.

Die verschiedenen erstellten Farbschemata können Abb. 6.2 auf Seite 36 entnommen werden.

Da das Finden der Akzentfarbe ein recht geradliniger Prozess ist fällt es schwer, dem Nutzer hier über die Wahl des von ihm präferierten Farbschemas hinaus viel Interaktivität zu bieten.

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

Grundfarbe



Komplementär



Triadisch (gespalten Komplementär)



Monochromatisch



ABBILDUNG 6.2: Beispielhaft errechnete Farbpaletten

ohne andere Farben zu wählen. Welche Grautöne genau genutzt werden muss dabei Subjektiv entschieden werden. Denkbar wären zum Beispiel #eeeeee und #666666.

[[Elizabeth\(\)](#)] 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 zurückgegriffen werden.

Hier sollte der Nutzer viele Freiheiten haben, das Tool sollte lediglich kontrollieren, ob die gewählten Grautöne innerhalb einiger Parameter liegen. So kann zum Beispiel über den *Lightness*-Wert des HSL-Farbraumes überprüft werden, ob sich zwei Grautöne deutlich genug unterscheiden. Ebenso kann über den *Saturation*-Wert sicher gestellt werden, dass die Grautöne nicht zu farbig sind.

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 auf diese Guidelines verweisen oder sie wiederholen, sondern eine interaktive Möglichkeit bieten, auch für diese Plattformen eine Farbpalette zu erstellen.

Die Android Guidelines geben zwar keine zwingenden Farben vor, jedoch bieten sie eine große Vorauswahl an Farben, die verwendet werden können. Alle diese Farben sind 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 vor allem für interaktive Elemente verwendet werden.

Die Empfehlung der Farben, basierend auf Adjektiven, kann für Android-Projekte weiterhin verwendet werden, jedoch mit einer limitierteren Farbauswahl. Akzentfarben müssen nicht mehr errechnet werden, hier kann dem Nutzer eine Auswahl der spezifizierten Akzentfarben gegeben werden, von der er eine auswählen kann.

Die iOS-Richtlinien sind deutlich weniger konkret. Unter iOS wird Farbe vor allem dafür genutzt, Interaktivität deutlich zu machen. Die guidelines schlagen hier 8 Farben vor, die auch dem Nutzer des Tools zur Auswahl gegeben werden sollten. Das Finden einer Akzentfarbe und von Grautönen entfällt dabei komplett. Das Vorschlagen der Farben auf Basis von Adjektiven wäre aber weiterhin, wenn auch recht limitiert, umsetzbar.

Kapitel 7

Bilder

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 beim Einsatz von Bildern 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 identifiziert 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 hinterlassen beim Betrachter einen unprofessionellen Eindruck. Zwar konnten keine wissenschaftlichen Belege dafür gefunden werden, ab welchem Grad der Verzerrung und wie schnell Menschen ein Bild als verzerrt erkennen, jedoch kann mit einiger Sicherheit davon ausgegangen werden, dass die meisten Menschen ein verzerrtes Bild erkennen, gerade wenn sich im Bild ihnen bekannte Objekte befinden.

Ursache für das Strecken oder Stauchen eines Bildes ist die Veränderung des ursprünglichen Seitenverhältnisses des Bildes beim Skalieren.

Als Seitenverhältnis wird das Verhältnis der beiden Kanten eines Bildes zueinander bezeichnet. Obwohl die Berechnung des Seitenverhältnisses an sich in der Praxis eine eher unwichtige

Rolle spielt (in der Regeln errechnen Programme dieses Verhältnis automatisch), wird sie für das Tool nötig sein, um eine korrekte Skalierung zu überprüfen. Wie bereits durch die Schreibweise (4:3, 16:9, ...) angedeutet, lässt sich das Seitenverhältnis berechnen, indem die Breite des Bildes durch die Höhe des Bildes dividiert wird.

Für ein Bild mit den Maßen $400px \times 300px$ ergibt sich also ein Seitenverhältnis von $400/300 = 1,33$ bzw. 4:3. Beim Skalieren des Bildes sollte also darauf geachtet werden, beide Kanten im gleichen Verhältnis zu skalieren, um das Bild nicht zu strecken oder zu stauchen (s. Abb. 7.1 auf Seite 39)



400px × 300px
4:3



200px × 150px
4:3



300px × 150px
2:1

ABBILDUNG 7.1: Beispiele für das korrekte und falsche Skalieren eines Bildes

Häufig treten Fehler in der Skalierung vor allem dann auf, wenn das Bild eine gegebene Fläche ausfüllen soll, die nicht dem Seitenverhältnis des Originalbildes entspricht. Ein reines Skalieren des Bildes reicht hier nicht aus, um ein befriedigendes Ergebnis zu erzielen, häufig müssen Teile des Bildes abgeschnitten werden, um die gewünschte Größe zu erreichen. Welcher Teil

des Bildes abgeschnitten wird und welcher erhalten bleibt, ist dabei wiederum eine ästhetische Entscheidung und kann nur in Abhängigkeit vom Bild und der gewünschten Wirkung getroffen werden. Im besten Falle wird die Kernaussage des Bildes durch die Beschneidung nicht verändert. Das Tool könnte dies beispielhaft mit einigen Bildern demonstrieren.

Grafikprogramme bieten für das Ausführen dieser Aufgabe verschiedenen Möglichkeiten, sowie das einfache Beschneiden des Bildes oder den Einsatz von Masken. Unter Umständen muss auf solche Programme zurück gegriffen werden, wenn das gewählte Textverarbeitungsprogramm oder der gewählte Editor diese Möglichkeit nicht bietet.

Auch im Web bieten sich hierfür verschiedene Möglichkeiten an. Die meiste Kontrolle über die Ausrichtung des beschnittenen Bildes bringt dabei die Verwendung des Bildes als Hintergrundbild über das CSS-Attribut *background-image* mit sich. Über den Wert *cover* des Attributes *background-size* wird sichergestellt, dass das Bild immer die gewünschte Fläche ausfüllt, ohne dabei gestreckt oder gestaucht zu werden (s. Quellcode 7.1 auf Seite 40).

```
1 .bg-image {  
2   width: 300;  
3   height: 150;  
4   background-image: url("image/landscape.jpg");  
5   background-size: cover;  
6 }
```

QUELLCODE 7.1: Verwendung eines Hintergrundbildes in CSS

7.2 Verpixelte Bilder

Neben gestreckten und gestauchten Bildern fallen auch verpixelte oder verschwommene Bilder unschön auf und verschlechtern die Qualität einer Gestaltung. Verpixelte Bilder können aus verschiedenen Gründen auftreten, die häufigste Ursache liegt aber in der Skalierung von Bildern über ihre Originalgröße hinaus. Da dieses Problem nur bei der Verwendung von Bitmaps auftritt, sei hier kurz angeschnitten, was genau beim Skalieren einer Bitmap geschieht.

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 et al.(2012)Enderle, Kansy, and Pfaff]

Bitmaps bestehen also aus einem Array von Pixeln, von denen jeder seine eigene Farbe und Position hat. Beim Skalieren einer Bitmap 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 Pixel zusammen gefasst werden und aufgrund des kleineren Bildes für das Menschliche Auge kein Unterschied in der Qualität zu erkennen ist.

Beim Skalieren nach oben müssen jedoch neue Pixel basierend auf der Datengrundlage des Bildes erzeugt werden. Bei der Skalierung nach oben können zwar, ähnlich wie bei der Skalierung nach unten, neue Pixel anhand der vorhandenen errechnet werden, jedoch wird diese Berechnung aufgrund der fehlenden Datengrundlage schnell ungenau und das Bild wirkt verschwommen. Nach Möglichkeit sollten Bilder also nicht über ihre Originalgröße hinaus skaliert werden..

Eine weitere Ursache für verpixelte Bilder liegt in der Verwendung von für das Medium unpassenden DPI-Werten. Da dieses Themengebiet sich aber weit von der Sicherung von gestalterischen Grundlagen entfernt soll es hier nicht behandelt werden.

Problematisch gestaltet sich eine interaktive Abbildung der gewonnen Erkenntnisse im Tool, da es sich hierbei eher um handwerkliche Richtlinien handelt, als um Regeln der Gestaltung, die durch einen *Trial & Error* Ansatz erlernt werden können.

Kapitel 8

Interaktive Elemente

Das letzte 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 möglich behandelt werden, um nicht zu sehr vom gesetzten Ziel des Projektes abzuweichen und den Fokus 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, der mit einer Maus bedient wird).

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, auch auf verschiedenen Plattformen, eingegangen.

8.1 Generell zu beachten

Laut den von Dix aufgestellten *Universal Design Principles* [Dix(2009), S.367] sollte ein interaktives System dem User so viel Feedback wie möglich geben. Dies gilt somit vor allem auch für die Elemente in diesem System, mit denen der Nutzer interagieren kann. Zu Feedback gehört

sowohl, dem Nutzer deutlich zu kommunizieren welche Elemente interaktiv sind, als auch dem Nutzer deutlich zu machen, dass eine Aktion erfolgt ist.

Für die meisten interaktiven Elemente gibt es einige Grundregeln und best practices, die aber individuell für eine Gestaltung angepasst werden können (beispielsweise durch die Anpassung der Farbe oder die Änderung der Grundform passend zur aktuellen Gestaltung). Je nach Element soll im Folgenden validiert werden, ob und in welchem Maße eine Anpassung des Elementes empfehlenswert ist.

Ein weiterer Faktor, der sich auf die Benutzbarkeit von interaktiven Elementen auswirkt, ist ihre Größe. Hier finden sich viele theoretische Grundlagen in der Mensch-Computer-Interaktion, letztendlich 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 wichtige Rolle. Während sich mit der Maus auch kleine Bereiche gezielt auswählen lassen, ist der menschliche Finger deutlich weniger präzise und die Fehlerrate steigt bei der Verwendung von kleineren Elementen deutlich an [[Park et al.\(2008\)](#)[Park, Han, Park, and Cho](#)].

Für die konkrete Größe von Touch-Elementen gibt es verschiedene Richtlinien von Unternehmen, deren Produktreihe Touch-Devices umfasst: Apple empfiehlt einen Touch-Bereich von mindestens 44pt mal 44pt [[Apple Inc.\(\)](#)], Google [[Google Inc.\(\)](#)] und Microsoft [[Microsoft Corporation\(\)](#)] sprechen von 48px. Zwar gibt es keinen definierten Standard für diese Größe, jedoch scheint eine Empfehlung im Bereich von 40px bis 50px als Ausgangspunkt valide zu sein.

8.2 Elemente

Im Folgenden soll auf eine Auswahl von interaktiven Elementen und deren Besonderheiten auf verschiedenen Plattformen eingegangen werden. Diese Liste besitzt keinerlei Anspruch auf Vollständigkeit, vielmehr wurde versucht sowohl häufig verwendete Elemente zu behandeln, als auch Elemente, die stellvertretend für ähnliche Elemente stehen können.

8.2.1 Buttons

Wie viele der interaktiven Elemente stellen auch 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. [[Dix\(2009\)](#)]

Ähnlich 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. Sie führen nur eine einzige Aktion aus, die durch einen Text und/oder ein Icon repräsentiert wird (auch hier lassen sich viele Diskussionen im Bereich der MCI darüber finden, in welcher Form die Aktion am Besten kommuniziert werden sollte).

Weiterhin haben sie einen Hintergrund der sich, häufig durch die Farbgebung, vom Hintergrund der Anwendung unterscheidet und reagieren auf verschiedene Arten von Interaktion, wie *hover* oder Betätigung. 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 in ihrer Größe unverändert. Weiterhin gibt es Versionen, bei denen der Hintergrund transparent ist und die Abgrenzungen des Buttons mit einem Rahmen markiert werden. Es bietet sich an, Buttons (und auch andere Elemente) mit der Akzentfarbe zu kennzeichnen, um sie aus der Gestaltung heraus stechen zu lassen.

8.2.1.1 Web

Im Web können Buttons verschieden definiert werden. Zum Einen gibt es ein explizites `<button>` Element, das einen Button im eigentlichen Sinne darstellt und beispielsweise zum absenden eines Formulars verwendet wird (`<input style="submit">` ist eine andere, aber ähnliche Variante). Diese Elemente werden von allen modernen Browsern bereits ohne manuelles Eingreifen wie Buttons gestaltet. Die Gestaltung ist dabei eher neutral und hängt vom verwendeten Betriebssystem und Browser ab, kann aber ohne Probleme angepasst werden.

Zum anderen werden *anchor tags* häufig visuell wie Buttons gestaltet.

Das Tool kann dem Nutzer an dieser Stelle einige Möglichkeiten bieten, einen Button zu erstellen und ihn warnen, sollte der Button beispielsweise zu klein sein. Dabei sollte auf die bereits bekannten Parameter wie z. B. die zuvor gewählte Farbe zurück gegriffen werden. Außerdem sollten dem Nutzer, je nach Wahl des HTML-Elementes, auch der zugehörige CSS-Code zugänglich gemacht werden.

8.2.1.2 Android

Die Material Design Guidelines spezifizieren drei unterschiedliche Arten von Buttons:

1. Floating Action Button
2. Raised Button

3. Flat Button

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

Floating Action Button

Pro Screen gibt es maximal einen Floating Action Button, er repräsentiert die auf dem Screen am häufigsten verwendete Aktion. Floating Action Buttons sollten nur *positive* Aktionen repräsentieren, zum Beispiel das Hinzufügen eines Elementes zu einer Liste. Der Button ist $56dp \times 56dp$ groß, rund, und mit einem Icon (in der Größe $24dp \times 24dp$) beschriftet. Er hat einen Schlagschatten (daher auch *Floating*). Als Farbe bietet sich die gewählte Akzentfarbe an. Bei einer Interaktion wird er dunkler und sein Schatten entfernt sich ein Stück nach unten (er schwebt nach der Metapher also höher). Mit den vom User zuvor ausgewählten Farben können der Normale und der Gedrückte Zustand des Buttons sehr simpel dargestellt werden. Abbildung 8.1 auf Seite 45 zeigt die Färbung am Beispiel der Akzentfarbe Pink (A200) auf. Im gepressten Zustand würde der Button entsprechend in einer dunkleren Abstufung (A400) gefärbt.



ABBILDUNG 8.1: Färbung eines Floating Action Buttons

Raised Button

Raised Buttons sollten genutzt werden, um Interaktivität auf einem Screen deutlich zu machen. Sie sollten immer dann genutzt werden, wenn der Anwendungsfall nicht für einen Floating Action Button oder Flat Buttons spricht. Raised Buttons sind mindestens 88dp breit, 36dp hoch, haben einen Corner-Radius von 2dp. Ihr Verhalten bei Interaktion ist ähnlich dem von Floating Action Buttons, sie werden dunkler und schweben höher. Sie lassen sich simultan zu den Floating Action Buttons färben, jedoch kann hier je nach Anwendungsfall unterschieden werden, ob die Akzentfarbe oder die Grundfarbe für den Button verwendet werden sollte.

Flat Button

Flat Buttons werden für weniger wichtige Aktionen und in Dialogen verwendet. Sie bestehen zunächst nur aus farbiger Schrift und haben keinen Hintergrund, bei Interaktion färbt sich ihr

Hintergrund jedoch. Flat Buttons werden in der Regel in der Grundfarbe dargestellt. Sie haben eine Höhe von 36dp, ein Touch Target von 48dp und eine Mindestweite von 88dp. Ihr Abstand nach innen und nach außen beträgt jeweils 8dp.

8.2.1.3 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. 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.

Die für das Tool interessantesten Buttons sind die System Buttons. Sie haben weder einen Rahmen noch einen Hintergrund und bestehen lediglich aus farbiger Schrift, wobei Rahmen oder Hintergrund bei Bedarf genutzt werden können. Die Textfarbe sollte hier die für die App gewählte Akzentfarbe sein.

8.2.2 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 der Text Input angesprochen werden, da sich die meisten Eigenschaften zwischen den verschiedenen Input Fields problemlos übertragen lassen. Das Tool sollte dem Benutzer jedoch nahelegen, die den Eingabedaten entsprechenden Input Fields zu verwenden.

8.2.2.1 Web

Im Web wird ein Text Input 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:

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

Alternativ kann statt dem Platzhaltertext auch ein Label außerhalb des Input Fields vergeben werden:

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

Interagiert der User mit dem Text Input, wird ein blinkender Cursor im Inneren platziert und der Rand des Elements färbt sich blau. Ähnlich wie auch Buttons hängt die Darstellung der 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. Das Tool sollte dem Nutzer hier, ähnlich wie bei den Buttons, die Möglichkeit geben, Input-Fields zu gestalten und ihn warnen, solle beispielsweise das Touch-Traget zu klein sein.

8.2.2.2 Android & iOS

Unter Android sind die Input Fields abstrakter und bestehen nur aus einem Label und einer Grundlinie. Bei Aktivierung floatet das Label nach oben, 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 Input 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 Inputs aus einem Label und einem Divider. Die Text Inputs passen sich visuell nicht an die Akzentfarbe an und können aus Xcode übernommen werden.

Das Tool kann dem Nutzer an dieser Stelle nicht viel Interaktivität bieten, es sollte ab dieser Stelle auf die entsprechenden Guidelines verweisen.

8.2.3 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, Select oder Checkboxes um beliebig viele aus einer Menge von 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 trotzdem eine gute Funktionalität auf allen Endgeräten sicher zu stellen, daher sollte sich das Tool für eine Verwendung des Standardverhaltens aussprechen.

8.2.4 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.
[Nielsen(2004)]

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

Kapitel 9

Fazit

Abschließend lässt sich die Zielsetzung als erreicht ansehen.

Es wurden relevante Themengebiete für die Konzeption eines Tools definiert. In diesen Themengebieten wurden Regeln festgehalten, die auch für eine spätere programmatische Umsetzung verwendet werden können. In manchen Gebieten, wie der Typographie oder den Farben lassen diese Regeln mehr Spielraum für Interaktivität zu, in Anderen, wie beispielsweise den Interaktiven Elementen und Bildern, weniger. Lediglich im Bereich Whitespace konnten keinerlei Regeln definiert werden.

Mögliche Ideen für eine Umsetzung wurden in Form von Wireframes festgehalten und im Bereich der Typographie konnte mit Hilfe eines *Proof of Concepts* gezeigt werden, dass auch eine tatsächliche technische Umsetzung mit moderatem Aufwand durchaus möglich ist.

Während des Projektes konnte aber auch festgestellt werden, dass das Übertragen von eher subjektiven Regeln für die Gestaltung von Artefakten in eine objektive, maschinenlesbare Form nicht einfach ist. Viele Entscheidungen, die Designer jeden Tag treffen, kann ein Computer nicht nachvollziehen. Auch können getroffene Entscheidungen nicht einfach als richtige oder falsche Referenzwerte übergeben werden, da deren Richtigkeit stark situationsabhängig ist.

Hier stellt sich die Frage, ob der Bereich *Machine Learning* eine sinnvolle Ergänzung für das Projekt bieten könnte, also ob ein Computer, ähnlich wie ein Mensch, ein *Gefühl* für eine gute oder schlechte Gestaltung entwickeln kann, wenn seine Datengrundlage groß genug ist.

Weiterhin ist auch die Frage nach dem tatsächlichen Nutzen des Tools noch ungeklärt. Zwar wird deutlich, dass ein solches Tool eine gewisse Relevanz hat und auch umgesetzt werden kann, es kann jedoch nicht festgestellt werden, ob das Tool dem Nutzer einen tatsächlichen Mehrwert liefern würde. Hierfür wären Nutzertests mit Prototypen nötig.

Literaturverzeichnis

- [W3C(a)] C21: Specifying line spacing in css. <http://web.archive.org/web/20160316091041/http://www.w3.org/TR/WCAG20-TECHS/C21.html>, a. Accessed: 29.06.2016.
- [css()] Css font stack: Web safe and web font family with html and css code. URL <http://www.cssfontstack.com/>.
- [Apple Inc.()] Apple Inc. Layout - visual design - ios human interface guidelines, 2016. URL <https://developer.apple.com/ios/human-interface-guidelines/visual-design/layout/>. Accessed 17.12.2016.
- [Baines and Haslam(2005)] Phil Baines and Andrew Haslam. *Type and typography*. Laurence King Publishing, 2005.
- [Berman(2010)] Margo Berman. *Street-smart advertising: How to win the battle of the buzz*. Rowman & Littlefield Publishers, 2010.
- [Bernard et al.(2002)Bernard, Fernandez, and Hull] Michael Bernard, Marissa Fernandez, and Spring Hull. The effects of line length on children and adults' online reading performance. *Usability News*, 4(2), 2002.
- [Campbell and Pisterman(1996)] Arnold Campbell and Susan Pisterman. A fitting approach to interactive service design: The importance of emotional needs. *Design Management Journal (Former Series)*, 7(4):10–14, 1996.
- [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.
- [Dix(2009)] Alan Dix. *Human-computer interaction*. Springer, 2009.

- [Dogusoy et al.(2016)Dogusoy, Cicek, and Cagiltay] Berrin Dogusoy, Filiz Cicek, and Kursat Cagiltay. How serif and sans serif typefaces influence reading on screen: An eye tracking study. In *International Conference of Design, User Experience, and Usability*, pages 578–586. Springer, 2016.
- [Elizabeth()] Laura Elizabeth. A simple web developer’s guide to color. URL <https://www.smashingmagazine.com/2016/04/web-developer-guide-color/>.
- [Enderle et al.(2012)Enderle, Kansy, and Pfaff] Günter Enderle, Klaus Kansy, and Günther Pfaff. *Computer Graphics Programming: GKS—The Graphics Standard*. Springer Science & Business Media, 2012.
- [Eysenck(1941)] Hans J Eysenck. A critical and experimental study of colour preferences. *The American Journal of Psychology*, 54(3):385–394, 1941.
- [Google Inc.()] Google Inc. Size tap targets appropriately, 2016. URL <https://developers.google.com/speed/docs/insights/SizeTapTargetsAppropriately>. Accessed 17.12.2016.
- [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/>.
- [Joblove and Greenberg(1978)] George H Joblove and Donald Greenberg. Color spaces for computer graphics. In *ACM siggraph computer graphics*, volume 12, pages 20–25. ACM, 1978.
- [Josephson(2008)] Sheree Josephson. Keeping your readers’ eyes on the screen: An eye-tracking study comparing sans serif and serif typefaces. *Visual communication quarterly*, 15(1-2):67–79, 2008.
- [Lehnert(2016)] Benedikt Lehnert. *Typoguide: A pocket guide to master every day’s typographic adventures (english edition)*, 2 2016. URL <http://amazon.de/o/ASIN/B01BW0A5ME/>.
- [Lindgaard et al.(2006)Lindgaard, Fernandes, Dudek, and Brown] Gitte Lindgaard, Gary Fernandes, Cathy Dudek, and Judith Brown. Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & information technology*, 25(2):115–126, 2006.
- [Livio(2003)] Mario Livio. *The golden ratio*. Aryeh Nir Publishers Limited, 2003.
- [Mayer(2005)] Horst Otto Mayer. *Einführung in die Wahrnehmungs-, Lern-und Werbe-Psychologie*. Oldenbourg Verlag, 2005.

- [Microsoft Corporation()] Microsoft Corporation. Guidelines for targeting, 2016. URL <https://msdn.microsoft.com/en-us/windows/uwp/input-and-devices/guidelines-for-targeting>. Accessed 17.12.2016.
- [Nickerson(1998)] Raymond S Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2):175, 1998.
- [Nielsen(2004)] Jakob Nielsen. Guidelines for visualizing links, 2004. URL <https://www.nngroup.com/articles/guidelines-for-visualizing-links/>. Accessed 17.12.2016.
- [Park et al.(2008)Park, Han, Park, and Cho] Yong S Park, Sung H Han, Jaehyun Park, and Youngseok Cho. Touch key design for target selection on a mobile phone. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 423–426. ACM, 2008.
- [Rider(2010)] Rose M Rider. Color psychology and graphic design applications. *Senior Honors Papers*, page 111, 2010.
- [Runk(2008)] Claudia Runk. *Grundkurs Typografie und Layout: Für Ausbildung und Praxis (Galileo Design)*. Galileo Design, 2 edition, 4 2008. ISBN 9783836212076. URL <http://amazon.de/o/ASIN/3836212072/>.
- [Strizver(2014)] Ilene Strizver. *Type Rules, Enhanced Edition: The Designer's Guide to Professional Typography*. John Wiley & Sons, 2014.
- [W3C(b)] W3C. G18: Ensuring that a contrast ratio of at least 4.5:1 exists between text (and images of text) and background behind the text, 2016b. URL <https://web.archive.org/web/20160914150636/http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/G18.html>. Accessed 18.12.2016.
- [Whelan(1994)] Bride Whelan. *Color harmony*. Rockport Publishers, 1994.
- [White(2011)] Alex W White. *The elements of graphic design: space, unity, page architecture, and type*. Skyhorse Publishing Inc., 2011.
- [Zhang et al.(2016)Zhang, Kothari, Butt, and Kumar] Jingxian Zhang, Neel Kothari, Asad Imtiaz Butt, and Ranjitha Kumar. What makes a brand look expensive? In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3263–3268. ACM, 2016.

Erklärung über die Selbständige Abfassung der Arbeit

Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Datum, Ort

Unterschrift