

Praxisprojekt
Titel

von

Student - 11098765

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

Betreuer: Prof. Dr. Christian Kohls

Technology, Arts and Sciences TH Köln

Technology, Arts and Sciences TH Köln, 23. November 2016

Abstract

Hier steht später das Abstract.

Inhaltsverzeichnis

Abstract	i
Abbildungsverzeichnis	iv
Quellcodeverzeichnis	iv
Abkürzungsverzeichnis	vi
1 Einleitung	1
1.1 Einleitung	1
1.2 Relevanz	1
1.3 Zielsetzung	2
1.4 Forschungsfrage	3
1.5 Vorgehensweise	3
2 Einleitung	5
2.1 Einleitung	5
2.2 Vorgehen	6
2.3 Kritische Reflexion	7
2.4 Mobile Plattformen	8

3	Einleitung	9
3.1	Schriftarten	9
3.1.1	Serif oder Serifenlos?	9
3.2	Schriftfamilien	10
3.2.1	Android	10
3.2.2	iOS	10
3.2.3	Mischen von Schriftfamilien	11
3.3	Schriftgrößen	11
3.3.1	Überschriften	11
3.3.1.1	Der goldene Schnitt	12
3.3.1.2	Typographic Scale	12
3.4	Abstände im Text	14
3.4.1	Zeilenhöhe und -länge	14
3.4.2	Absätze und Überschriften	14
3.5	Kontrast	15
3.5.1	Berechnung	15
4	Einleitung	16
4.1	Herleitung	16
4.2	Grid Systems im Web	16
4.2.1	Abbildung im Tool	17
4.3	Grid Systems auf nativen Plattformen	18
4.3.1	iOS	18

4.3.2	Android	18
4.4	Grid Systems im Print	19
4.4.1	Berechnung	19
4.4.2	Abbildung im Tool	20
5	Fazit	21
5.1	Zusammenfassung	21
5.2	Fazit	21
5.3	Schlussbemerkung	21

Abbildungsverzeichnis

Quellcodeverzeichnis

Abkürzungsverzeichnis

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

Kapitel 1

Einleitung

1.1 Einleitung

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

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

1.2 Relevanz

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

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

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

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

[Raymond S Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. Review of general psychology, 2(2):175, 1998.]

Menschen verwenden Informationen also so, dass sie als Argumente für ihre bisherige Meinung genutzt werden können und entwerfen 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.

[Arnold Campbell and Susan Pisterman. A fitting approach to interactive design.]

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

[Gitte Lindgaard, Gary Fernandes, Cathy]

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

1.3 Zielsetzung

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

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

1.4 Forschungsfrage

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

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

1.5 Vorgehensweise

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

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

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

Kapitel 2

Einleitung

2.1 Einleitung

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

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

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

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

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

2.2 Vorgehen

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

Diese wurden zur Orientierung zunächst grob in vier Kategorien unterteilt:

* Präsentation * Plakat * Interaktiv * Text

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

Die genauen Zahlen der jeweiligen Kategorien können *Tabelle 1* entnommen werden. Bei der Betrachtung der Werte fällt auf, dass nahezu 50

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

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

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

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

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

2.3 Kritische Reflexion

Auch wenn dieses fehlerorientierte Vorgehen für das Projekt sehr hilfreich ist, bringt es einige Gefahren mit sich.

So ist das Material von nur zwei Semestern keineswegs dazu geeignet, empirische Ergebnisse zu liefern. Die erarbeiteten Ergebnisse liefern lediglich einen Überblick über die Fehler, die in naher Vergangenheit vorherrschend waren. Außerdem ändern sich Inhalte von Modulen häufig

von Semester zu Semester, eine Untersuchung im nächsten Jahr würde also mit hoher Wahrscheinlichkeit neue Fehler zeigen.

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

2.4 Mobile Plattformen

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

Kapitel 3

Einleitung

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

3.1 Schriftarten

Um einen Text zu setzen müssen eine Reihe von Entscheidungen getroffen werden. Die erste dieser Entscheidungen stellt die Entscheidung über die Schriftart dar. Schriftarten werden in Kategorien unterteilt, die nicht immer einheitlich sind. So verwendet Strizver [Strizver, Ilene.TypeRules,EnhancedEdition]

3.1.1 Serif oder Serifenlos?

Je nach Medium lassen sich in der Lesegeschwindigkeit bei serifen und serifenlosen Schriftarten Unterschiede feststellen, so kommen sowohl Josephson [Josephson,S.(2008).Keepingyourreaders'eyesonthescreen:An

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

3.2 Schriftfamilien

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

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

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

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

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

3.2.1 Android

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

3.2.2 iOS

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

3.2.3 Mischen von Schriftfamilien

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

3.3 Schriftgrößen

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

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

3.3.1 Überschriften

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

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

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

3.3.1.1 Der goldene Schnitt

Der goldene Schnitt findet in 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, M. (2003). The golden ratio (p. 4). Aryeh Nir Publishers Limited.] wird dieses Verhältnis durch eine unendliche, sich nie

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

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

3.3.1.2 Typographic Scale

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

[HIERBILDVONSCALEAUSBUCH]

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

[HIERERSTESBILDVONGESETZTEMTEXT]

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

[HIERZWEITESBILDVONGESETZTEMTEXT]

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

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

“ 14px * 1,62 = 22,68px => 24px 16px * 1,62 = 25,92px => 24px 18px * 1,62 = 29,16px => 24px 21px * 1,62 = 34,02px => 36px “

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

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

“ 14px * 1,62 = 22,68px => 24px 16px * (1,62 + 0,2) = 29,12px => 24px 18px * (1,62 + 0,4) = 36,36px => 36px 21px * (1,62 + 0,7) = 48,72px => 48px “

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

3.4 Abstände im Text

TBD

3.4.1 Zeilenhöhe und -länge

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

3.4.2 Absätze und Überschriften

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

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

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

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

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

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

[*HIERNOCHBILD*]

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

3.5 Kontrast

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

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

3.5.1 Berechnung

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

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

Kapitel 4

Einleitung

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

4.1 Herleitung

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

4.2 Grid Systems im Web

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

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

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

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

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

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

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

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

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

4.2.1 **Abbildung im Tool**

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

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

4.3 Grid Systems auf nativen Plattformen

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

4.3.1 iOS

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

4.3.2 Android

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

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

4.4 Grid Systems im Print

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

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

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

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


4.4.1 Berechnung

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

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

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

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

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

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

4.4.2 Abbildung im Tool

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

Kapitel 5

Fazit

5.1 Zusammenfassung

5.2 Fazit

5.3 Schlussbemerkung