



TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Fakultät Informatik

Analyse und Überarbeitung des Graphical User Interfaces von EB GUIDE Studio 6 zur Steigerung der Usability

Bachelorarbeit im Studiengang Medieninformatik

vorgelegt von

Sandra Schumann

Matrikelnummer 302 0357

Erstgutachter: Prof. Dr. Korbinian Riedhammer

Zweitgutachter: Prof. Dr. Matthias Teßmann

© 2019

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Prüfungsrechtliche Erklärung der/des Studierenden

Angaben des bzw. der Studierenden:

Name: _____ Vorname: _____ Matrikel-Nr.: _____

Fakultät: _____ Studiengang: _____

Semester: _____

Titel der Abschlussarbeit:

Ich versichere, dass ich die Arbeit selbständig verfasst, nicht anderweitig für Prüfungszwecke vorgelegt, alle benutzten Quellen und Hilfsmittel angegeben sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Ort, Datum, Unterschrift Studierende/Studierender

Erklärung zur Veröffentlichung der vorstehend bezeichneten Abschlussarbeit

Die Entscheidung über die vollständige oder auszugsweise Veröffentlichung der Abschlussarbeit liegt grundsätzlich erst einmal allein in der Zuständigkeit der/des studentischen Verfasserin/Verfassers. Nach dem Urheberrechtsgesetz (UrhG) erwirbt die Verfasserin/der Verfasser einer Abschlussarbeit mit Anfertigung ihrer/seiner Arbeit das alleinige Urheberrecht und grundsätzlich auch die hieraus resultierenden Nutzungsrechte wie z.B. Erstveröffentlichung (§ 12 UrhG), Verbreitung (§ 17 UrhG), Vervielfältigung (§ 16 UrhG), Online-Nutzung usw., also alle Rechte, die die nicht-kommerzielle oder kommerzielle Verwertung betreffen.

Die Hochschule und deren Beschäftigte werden Abschlussarbeiten oder Teile davon nicht ohne Zustimmung der/des studentischen Verfasserin/Verfassers veröffentlichen, insbesondere nicht öffentlich zugänglich in die Bibliothek der Hochschule einstellen.

Hiermit ☐ genehmige ich, wenn und soweit keine entgegenstehenden
Vereinbarungen mit Dritten getroffen worden sind,
☐ genehmige ich nicht,

dass die oben genannte Abschlussarbeit durch die Technische Hochschule Nürnberg Georg Simon Ohm, ggf. nach Ablauf einer mittels eines auf der Abschlussarbeit aufgebrachten Sperrvermerks kenntlich gemachten Sperrfrist

von _____ Jahren (0 - 5 Jahren ab Datum der Abgabe der Arbeit),

der Öffentlichkeit zugänglich gemacht wird. Im Falle der Genehmigung erfolgt diese unwiderruflich; hierzu wird der Abschlussarbeit ein Exemplar im digitalisierten PDF-Format auf einem Datenträger beigelegt. Bestimmungen der jeweils geltenden Studien- und Prüfungsordnung über Art und Umfang der im Rahmen der Arbeit abzugebenden Exemplare und Materialien werden hierdurch nicht berührt.

Ort, Datum, Unterschrift Studierende/Studierender

Formular drucken

Kurzdarstellung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Abstract

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Elektrobit Automotive GmbH	1
1.2. Abteilung User Experience	1
1.3. Motivation	2
1.4. Zielsetzung	2
1.5. Aufbau der Arbeit	2
2. Theorie	3
2.1. Grafische Benutzeroberfläche	3
2.2. Ergonomie	3
2.3. Usability	3
2.4. User Experience	4
2.5. Human - Centered - Design - Process	5
2.6. Gestaltprinzipien der Usability	6
2.7. EB Guide Studio	10
3. Analysen	13
3.1. Voruntersuchungen	13
3.1.1. Ausgangssituation	13
3.1.2. Vorgehensweise	13
3.1.3. Ergebnisse	13
3.2. Verbesserungen	17
3.2.1. Auswahlkriterien	17
3.2.2. Gewinn für den Nutzer	17
3.2.3. Design der Verbesserungen	17
4. Umsetzung	21
4.1. Prototyp Multiselektion	21
4.1.1. Zielsetzung	21
4.1.2. Axure RP	21
4.1.3. Interaktionsmöglichkeiten	21
4.1.4. Vorgehensweise	21
4.2. Implementierung Filter	21
4.2.1. Zielsetzung	21

4.2.2. Projektaufbau	21
4.2.3. Vorgehensweise	21
5. Usability - Test	23
5.1. Lookback	23
5.2. Remote Usability - Test	23
5.3. Arbeitsaufgaben	23
5.4. Ergebnisse altes Interface	23
5.5. Ergebnisse überarbeitetes Interface	23
5.6. Vergleich	23
6. Fazit	25
6.1. Ergebnisse	25
6.2. Ausblick	25
A. Supplemental Information	27
Abbildungsverzeichnis	29
Tabellenverzeichnis	31
List of Listings	33
Literaturverzeichnis	35

Kapitel 1.

Einleitung

In diesem einführenden Kapitel wird zunächst kurz das Partnerunternehmen der Abschlussarbeit mit der zugehörigen Abteilung User Experience vorgestellt. Weiterhin wird die Motivation dieser Arbeit und das verfolgte Ziel erläutert, bevor es noch einen Überblick über die folgenden Kapitel gibt.

1.1. Elektrobit Automotive GmbH

Partner der Bachelorarbeit ist die Firma Elektrobit Automotive GmbH - im Folgenden nur noch als EB bezeichnet. EB ist ein vielfach ausgezeichnetes, internationales Unternehmen, welches sich auf die Entwicklung von Produkten und Dienstleistungen im Bereich der Automobilindustrie spezialisiert hat. Mit über 30 Jahren Branchenerfahrung bietet EB seinen Kunden unter anderem innovative Lösungen für das vernetzte Fahrzeug, Human Machine Interface Technologien (HMI), Navigations- und Fahrassistenzsysteme und Steuergeräte. Die Automotive Software von EB befindet sich in über 1 Billionen Geräten die in mehr als 90 Millionen Fahrzeugen weltweit Verwendung finden. Mit über 2300 beschäftigten Mitarbeitern, verteilt auf 3 Kontinente und 9 Länder, und einer durchschnittlichen jährlichen Wachstumsrate von über 10 % ist EB ein weltweit etabliertes Unternehmen mit Hauptsitz in Erlangen[\[Eleka\]](#).

1.2. Abteilung User Experience

Jedes Gerät das für den alltäglichen Gebrauch gedacht ist, sollte eine erfolgreiche Interaktion gewährleisten. Dafür ist eine Schnittstelle zwischen Mensch und Maschine (HMI) die einen intuitiven, einfachen und schnellen Umgang mit diesem Gerät ermöglicht unabdingbar. Um die Erfahrungsqualität im Allgemeinen möglichst hoch zu halten wünschen Nutzer sich auf ihre Bedürfnisse angepasste User-Interfaces in allen Lebensbereichen, womit der Bereich UX auch in der Automobilbranche einen hohen Bedeutungsgrad genießt. Die Abteilung User Experience von EB befasst sich vor allem mit der Entwicklung multimodaler HMIs

für Kombiinstrumente, Head Units und Head-Up Displays. Diese werden von EB von der Konzeptphase bis hin zur Serienentwicklung mit Hilfe von EB-GUIDE entwickelt.

1.3. Motivation

TODO

1.4. Zielsetzung

Ziel dieser Bachelorarbeit ist es, Schwachstellen im User Interface von EB GUIDE zu identifizieren und durch deren Verbesserung die Usability von EB GUIDE zu erhöhen. Dafür ist zuerst eine Analyse der Arbeitsabläufe innerhalb der Modellierungsarbeit nötig um entsprechende Probleme im User Interface zu erkennen. Anschließend gilt es Konzepte zu entwickeln welche, durch Anpassung und Überarbeitung der entsprechenden Komponenten in der Benutzerschnittstelle, diese Probleme beheben oder minimieren. Diese Konzepte werden anschließend noch grafisch und interaktiv visualisiert, um abschließende Usability-Tests durchführen zu können. Dabei wird eine identische Aufgabenstellung von Probanden mit dem Alten und dem Überarbeiteten Interface durchgeführt und durch den Vergleich der Ergebnisse festgestellt, ob die Anpassungen die Usability von EB Guide erhöht haben.

1.5. Aufbau der Arbeit

TODO

Kapitel 2.

Theorie

2.1. Grafische Benutzeroberfläche

Der Begriff Benutzerschnittstelle bezeichnet alle Komponenten eines interaktiven Systems, die dem Benutzer Interaktionsmöglichkeiten mit selbigem System bieten um ein verfolgtes Ziel zu erreichen. Die grafische Benutzeroberfläche (GUI) bezeichnet hierbei den sichtbaren Anteil des Systems und damit nur einen Teil der gesamten Benutzerschnittstelle, zu der auch nicht sichtbare Teile wie z.B. die Funktionslogik gehören[Saro 16]. Heutzutage sind die meisten Benutzeroberflächen auch grafische Benutzeroberflächen, mit denen in den häufigsten Fällen die Interaktion mit dem Nutzer über direkte Manipulation stattfindet[Niel 95].

2.2. Ergonomie

Unter Ergonomie versteht man im Allgemeinen die "Lehre von der menschlichen Arbeit und die Erkenntnis ihrer Gesetzmäßigkeiten"[Ergo 14]. Hierbei ist es wichtig zu verstehen, dass dabei der Fokus nicht ausschließlich auf einer technischen Komponente liegt, sondern das Zusammenspiel von Mensch, der zugeteilten Aufgabe und den verfügbaren Werkzeugen betrachtet wird[Saro 16]. Im Bezug auf Software bedeutet Ergonomie also konkret diese gut handhabbar und benutzerorientiert zu gestalten.

2.3. Usability

Mit immer höherer Komplexität von Systemen und Anwendungen kam der Begriff und das Verlangen nach "Benutzerfreundlichkeit" auf. Dieser Begriff suggeriert das lediglich die einfache Benutzung eines Systems ausschlaggebend ist, vernachlässigt hierbei jedoch die Notwendigkeit den Nutzer beim Erreichen seiner Ziele passend zu unterstützen. Dies ist auch der Grund dafür das bald, statt auf "Benutzerfreundlichkeit" auf "Gebrauchstauglichkeit"(engl. Usability) geachtet wurde. Im Gegensatz zur Ergonomie handelt es sich bei Usability nicht

um eine eigenständige wissenschaftliche Disziplin, sondern um eine qualitative Anforderung an ein System[Saro 16]. Konkret spricht man bei einer Software-Anwendung von einer hohen Usability, wenn sie von der für sie bestimmten Zielgruppe effizient verwendet werden kann, also das verfolgte Ziel zufriedenstellend erreicht wird[Rich 16]. Hierfür ist es entscheidend sich bewusst zu machen das ein technisches System oder Software immer Teil eines großen Handlungsablaufes ist und dazu dient Schritte dieses Handlungsablaufes zu erledigen. Deshalb muss das System den Anforderungen dieses Ablaufes entsprechen und darf während der Entwicklung nicht getrennt davon betrachtet werden[Saro 16].

2.4. User Experience

Entgegen einer häufigen Annahme bezeichnen Usability und User Experience (UX) nicht das Gleiche. Usability bezeichnet lediglich ein Teil der gesamten User Experience eines Systems[Knig 19]. UX bezieht sich nicht nur auf die reine Nutzungszeit eines Systems, sondern berücksichtigt auch den Zeitraum davor und danach, bezeichnet als Antizipierte Nutzung und Verarbeitung der Nutzungssituation. Usability ist hierbei, wie in Abb. 2.1 zu sehen, als wichtiger Faktor der User Experience in der aktiven Nutzungsphase zu betrachten, jedoch nicht mit dem Begriff gleichzusetzen [Saro 16]. Durch die zusätzliche Betrachtung der Effekte auf den Nutzer vor und nach der Nutzung, wie beispielsweise Erwartungen an das Produkt und Akzeptanz des selbigen, entstehen hier auch Verbindungen zur Gestaltung der Benutzerschnittstelle und dem Produkt-Design[Rich 16]. Zusammenfassend lässt sich festhalten das Usability zwar die funktionsbezogene Betrachtungsweise abdeckt, die User Experience als ganzes jedoch auch emotionale Faktoren bezüglich Design und Ästhetik berücksichtigt um das Nutzungsvergnügen möglichst hoch zu halten. Zusätzlich ist eine gute User Experience notwendig wenn ein Produkt auf dem Markt bestehen will. Sobald es mehr als ein Produkt zur Lösung der gleichen Aufgabenstellung gibt, wird das mit der besseren User Experience Verwendung finden[Knig 19].

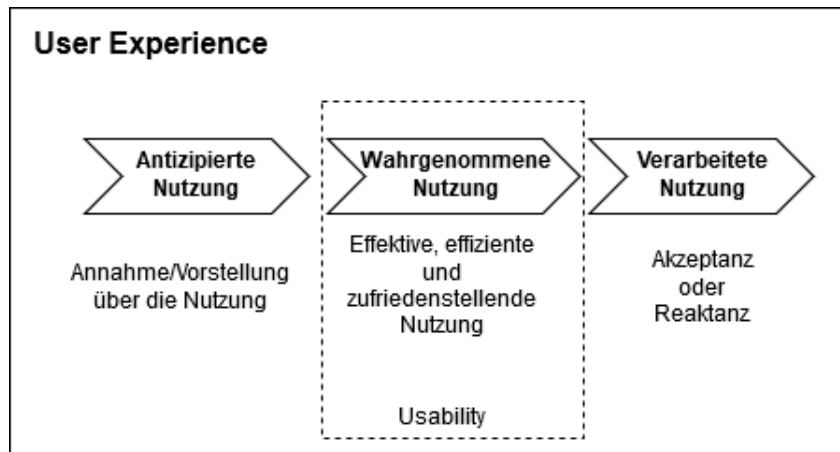


Abbildung 2.1.: Zusammenhang Usability und User Experience (nach [Saro 16])

2.5. Human - Centered - Design - Process

TODO

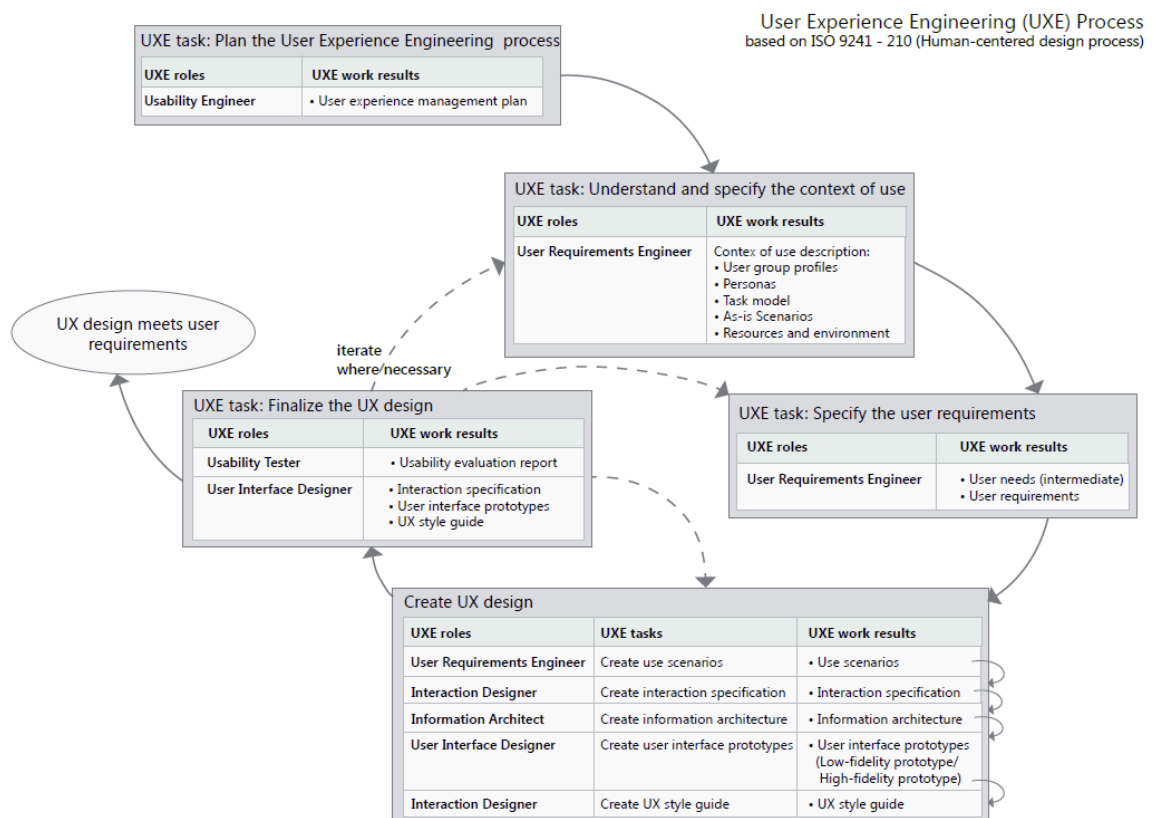


Abbildung 2.2.: Human - Centered - Design - Process bei Elektrobit

2.6. Gestaltprinzipien der Usability

TODO Ausformulieren

[[Norm 16](#)] Für Einleitung etwas suchen

Konsistenz

Konsistenz im Design wirkt sich insofern positiv auf die Usability eines Systems aus, das die Benutzbarkeit steigt wenn gleiche Funktionen in einem System auch auf identische Art und Weise dargestellt werden. Dadurch wird es dem Nutzer ermöglicht bereits Gelerntes in einem neuen Kontext anzuwenden, ohne effektiv darüber nachdenken zu müssen, neue Dinge schneller zu lernen und sich auf die relevanten Dinge einer Aufgabe zu konzentrieren. Konsistenz in Systemen lässt sich in vier Kategorien unterteilen, die im Folgenden jeweils kurz erläutert werden.

Ästhetische Konsistenz Ästhetische Konsistenz bezieht sich auf Konsistenz in Stil und Aussehen. Dadurch entsteht ein hoher Wiedererkennungswert und es wird Zugehörigkeit signalisiert. Das einfachste Beispiel hierfür ist ein Firmenlogo, welches immer mit konsistenter Schriftart und Farbgebung auftaucht.

Funktionale Konsistenz Funktionale Konsistenz bezieht sich auf die Konsistenz von Bedeutung und Aktion, beispielsweise zeigt eine Ampel immer orange bevor sie rot wird. Der Nutzer kann also sicher mit einer immer gleichen darauffolgenden Aktion auf den aktuellen Zustand rechnen. Die Usability und Lernfähigkeit wird durch funktionale Konsistenz erhöht, indem diese bereits verinnerlichten Abläufe einfach auf neue Situationen übertragen werden können. So wird beispielsweise ein einheitlicher Playbutton für alle Geräte, von Kassettenrekorder bis Streamingdienst, verwendet. Die Nutzung bereits bekannter Symbole ermöglicht dem Nutzer also eine intuitive Interaktion und macht es möglich die Aufmerksamkeit auf unbekannte Aspekte zu richten, die tatsächlich noch neu gelernt werden müssen.

Interne Konsistenz Interne Konsistenz bezieht sich auf Konsistenz mit anderen Elementen im System, beispielsweise sind Wegweiser in einem Park konsistent zueinander. Dies erzeugt ein Vertrauensgefühl bei den Nutzern, und vermittelt ein durchdachtes Designkonzept und erweckt nicht den Eindruck das alle Komponenten einfach zusammengewürfelt wurden.

Externe Konsistenz Externe Konsistenz bezieht sich auf Konsistenz mit anderen Elementen in der Umgebung. Die Vorteile der Internen Konsistenz werden hierbei systemübergreifend erweitert. Das Erreichen externer Konsistenz gestaltet sich jedoch auch schwieriger, da unabhängige Systeme selten exakt gleiche Designstandards haben.

Nicht alle dieser Konsistenzstandards können oder sollten in allen Fällen angewendet werden. Es lässt sich jedoch festhalten, dass Elemente in einer logischen Gruppe immer ästhetisch und funktional konsistent sein sollten. Insgesamt sollte ästhetische und funktionale Konsistenz, wann immer möglich berücksichtigt werden, da durch ästhetische Konsistenz bei der Einführung einmaliger Identitäten Wiedererkennung gewährleistet werden kann und funktionale Konsistenz aus den bereits genannten Gründen die Usability eines Systems erhöht. Abschließend sollten Systeme immer eine interne Konsistenz aufweisen, sollten bereits Designstandards existieren, gilt es diese zu analysieren.[\[Lidw 10\]](#)

Sichtbarkeit

Das Prinzip der Sichtbarkeit besagt, dass Systeme benutzbarer sind, wenn der aktuelle Status des Systems, mögliche Aktionen und deren Auswirkungen für den Nutzer deutlich erkennbar sind. Es beruht auf der Erkenntnis, dass Menschen schneller und besser Lösungswege finden, wenn sie aus einer Reihe von Optionen auswählen können, anstatt sich selbstständig an alle Möglichkeiten aus dem Stegreif erinnern zu müssen[\[Lidw 10\]](#). Das macht es zu dem wichtigsten Prinzip für hohe Usability in komplexen Systemen, jedoch auch zu dem, welches am meisten verletzt wird[\[Norm 16\]](#). Häufig wird versucht, alle möglichen Optionen, die ein System bietet, sichtbar zu machen, was vor allem in komplexeren Systemen dazu führt, dass sie relevanten Optionen durch Informationsüberladung auf Seiten des Nutzers schwerer zu erreichen sind. Anstatt die Usability zu erhöhen, wird also genau das Gegenteil erzielt. Lösungen für diese Problemstellung sind beispielsweise eine hierarchische Anordnung der Elemente oder Kontextsensitivität des Systems. Bei der hierarchischen Anordnung werden Funktionen und Informationen in logische Kategorien unterteilt und in übergeordneten Menüs versteckt, welche bei Bedarf ausgeklappt werden können. Ein Beispiel hierfür ist ein Dropdownmenü, wie man es von vielen Websites kennt. In einem kontextsensitiven System werden Aktionsmöglichkeiten und Informationen je nach aktuellem Status des Systems versteckt oder angezeigt. Beispielsweise bekommt man in einer Modellierungssoftware mit unterschiedlichen Elementen immer nur die Properties angezeigt, die für das aktuell ausgewählte Element relevant sind, und nicht alle Properties, die im gesamten System existieren[\[Lidw 10\]](#).

Affordanz

Affordanz bezeichnet die Möglichkeiten, mit einem Objekt zu interagieren. Beispielsweise kann eine Checkbox an und wieder abgewählt oder ein Schieberegler nach oben und unten

geschoben werden. Die sichtbare Affordanz bezeichnet die Eigenschaft das einem Objekt die Interaktionsmöglichkeiten bereits angesehen werden können, ohne mit ihm interagiert zu haben. Sichtbare Affordanz ist vor allem in der User Interface Gestaltung wichtig, weil praktisch betrachtet alle Pixel auf einem Bildschirm anklickbar sind, jedoch in den meisten Fällen keine Aktion durch das klicken ausgelöst wird. Deshalb ist es wichtig dem Nutzer durch das Aussehen der Element zu vermitteln ob diese, wenn sie geklickt werden eine Aktion auslösen, um dem Nutzer wahlloses Klicken durch das Interface zu ersparen, bis ein interaktives Objekt gefunden wird. Solche Probleme können visuell gelöst werden, indem man Objekte im User Interface wie Objekte in der echten Welt aussehen lässt, also beispielsweise einen Button dreidimensional gestaltet. Alternativ kann man beispielsweise alle anklickbaren Objekte etwas anders gestalten als den Rest der Objekte im Interface, um dem Nutzer zu vermitteln das hier eine Interaktion stattfinden kann[[Knig 19](#)].

Rückmeldung

Eine der grundlegendsten Richtlinien um die Usability eines Systems zu erhöhen, ist es dem Nutzer immer eine Rückmeldung auf seine Aktionen zu geben. Das bedeutet dem Nutzer immer den aktuellen Systemstatus anzuzeigen und wie seine Aktion vom System interpretiert wurde. Eine Rückmeldung ist vor allem auch dann wichtig, wenn die gewünschte Aktion nicht erfolgreich vom System ausgeführt wurde. Durch fehlende Rückmeldungen kommt Misstrauen beim Nutzer auf, weil ihm nicht vermittelt wird ob auf seine Aktionen auch eine Reaktion des Systems erfolgt[[Knig 19](#)]. Dieses Problem tritt beispielsweise auf, wenn ein System lange braucht um eine Eingabe zu verarbeiten und es versäumt dies dem Nutzer mitzuteilen. So ein Verhalten könnte zur fälschlichen Annahme führen das System wäre kaputt, oder dazu das der Nutzer beginnt neue Interaktionsmöglichkeiten anzuklicken. Reagiert das System innerhalb 0.1 Sekunden, nimmt der Nutzer dies als sofortiges Feedback wahr und keine gesonderte Meldung ist nötig. Bewegt sich die Reaktionszeit zwischen 0.1 und 1.0 Sekunden ist führt gewöhnlich auch keine besondere Rückmeldung vom System nötig, auch wenn der Nutzer hier bereits nicht mehr das Gefühl hat direkt mit den Daten zu interagieren, da eine kurze Verzögerung stattfindet. Sobald die Wartezeit auf Seiten des Nutzer jedoch eine Sekunde überschreitet sollte sich der Cursor in eine Sanduhr oder ähnliches verwandeln und damit die Rückmeldung geben, dass das System beschäftigt ist. Sollte die systemseitige Verarbeitung der Eingabe länger als 10 Sekunden dauern ist es ratsam die Ladeanzeige durch eine konkrete Fortschrittsleiste zu ersetzen[[Niel 95](#)].

Mapping

Mapping bezeichnet die Beziehung zwischen Bedienelementen und den Effekten den deren Aktivierung auslöst. Wenn der Effekt den eine Nutzerinteraktion nach sich zieht, den Er-

wartungen des Nutzers entspricht, handelt es sich um gutes Mapping. Die ist beispielsweise bei einem elektronischen Fensterheber der Fall. Wird hier der Hebel nach oben bewegt, hebt sich das Fenster, bewegt man den Hebel nach unten senkt es sich. Gutes Mapping wird zum Großteil durch Ähnlichkeit, das Verhalten oder der Bedeutung innerhalb eines Layouts erreicht. Entspricht beispielsweise das Layout von Herdplattenreglern der Anordnung der Platten wird gutes Mapping durch das Layout erzeugt, der bereits beschriebene Fensterheber arbeitet mit dem Verhalten, und die Tatsache das ein Notfallknopf rot eingefärbt wird ist darauf zurückzuführen das die meisten Menschen rot mit Gefahr oder dem Stopplicht einer Ampel assoziieren. In jedem dieser Fälle macht die Ähnlichkeit es möglich den Effekt der Handlung vorherzusehen, und vereinfacht dadurch die Bedienung für den Nutzer. Aktionsmöglichkeiten müssen so platziert werden das ihre Position und ihr Verhalten dem Layout und dem Verhalten der Anwendung angepasst sind. Außerdem sollte es vermieden werden durch eine identische Aktion verschiedene Reaktionen auszulösen[Lidw 10].

Einschränkungen

Einschränkungen in einem System limitieren die Interaktionsmöglichkeiten für den Nutzer. Wird beispielsweise ein Button ausgegraut, der im aktuellen Kontext ohnehin keine Aktion ausführen würde wird der Nutzer rein optisch daran gehindert eine nicht zielführende Aktion auszuführen. Durchdachte Einschränkungen dieser Art machen ein Design einfacher nutzbar und reduziert deutlich die Wahrscheinlichkeit von Fehlschlägen während der Systeminteraktion [Lidw 10].

Physische Einschränkungen Physische Einschränkungen limitieren den Bereich in dem Aktionen ausgeführt werden können, indem sie Eingaben des Nutzers umwandeln oder umleiten. Eine Art der Einschränkungen ist das Konvertieren der Eingabe in lineare oder kurvenförmige Bewegung, wie es beispielsweise bei der Interaktion mit einer Scrollbar der Fall ist. Mithilfe von Achsen können wirkende Kräfte in Rotationsbewegungen umgewandelt werden, was eine Kontrolloberfläche mit unendlicher Größe auf einem kleinen Feld erzeugt, was am Beispiel der Computermaus gut zu erkennen ist. Die Letzte Form der Einschränkung passiert über Barrieren, die die Eingabe verlangsamen, komplett ausbremsen oder umleiten. Die Einfassung eines Computerbildschirms beschränkt beispielsweise physisch die Interaktionsfläche für den Nutzer. Allgemein sind Physische Einschränkungen nützlich um die Anzahl fehlerhafter Eingaben zu vermeiden, oder manche Eingaben erst gar nicht zu ermöglichen[Norm 16].

Psychologische Einschränkungen Psychologische Einschränkungen limitieren die Anzahl möglicher Aktionen durch Nutzung des Wissens über das Verhalten und die Denkweise der Nutzer. Dies passiert durch den Einsatz von Symbolen, nutzen bekannter Konventionen

oder durch das bereits erwähnte Mapping. Symbole sind sinnvoll um Dinge zu benennen, zu erklären oder auch um Warnungen visuell, auditiv oder fühlbar darzustellen. Mit Konventionen macht man sich bekannte Interaktionsmöglichkeiten zunutze, weshalb sie sich gut eignen Systeme sowohl konsistent als auch leicht benutzbar zu machen. Mappings sind, aus bereits erwähnten Gründen, nützliche um dem Benutzer zu vermitteln welche Aktionen aufgrund der Sichtbarkeit, Position oder dem Aussehen der Elemente möglich sind[[Norm 16](#)].

Zusammenfassend sollten Einschränkungen im Allgemeinen verwendet werden um die Benutzbarkeit eines Systems zu vereinfachen und die Anzahl an Fehlern zu minimieren. Physische Einschränkungen dienen hierbei eher dem Verhindern ungewollter Eingaben oder gefährlicher Aktionen, Psychologische Einschränkungen sollen das Design eines Systems für den Nutzer klarer und intuitiver gestalten.

Vorteile für den Nutzer

Durch das Befolgen der soeben erläuterten Gestaltprinzipien bei dem Entwurf eines User Interfaces, ist es möglich sich die Art und Weise wie Nutzer visuelle Reize verarbeiten zunutze zu machen um die Benutzerfreundlichkeit des Systems zu erhöhen. Das passiert vor allem dadurch, dass man die kognitive Last des Nutzers verringern während er sich mit dem Interface auseinander setzt, indem man bereits bekannten Gestaltprinzipien nutzt. Das bedeutet, dass der Nutzer seine Energie nicht darauf verschwenden muss darüber nachzudenken wie mit den Bestandteilen des Interfaces interagiert werden kann, oder was deren Funktionen sein könnten[[Knig 19](#)].

2.7. EB Guide Studio

Wie bereits in Abschnitt 1.2 erwähnt dient EB GUIDE der Entwicklung multimodaler HMIs. Um nicht nur das Design sondern auch das Verhalten von User Interfaces bestimmen zu können und eine Auslieferung auf das Zielsystem zu ermöglichen besteht die Produktlinie EB GUIDE aus den verschiedenen, in Abb. 2.3 zu sehenden, Komponenten. Hierbei wird zwischen Komponenten für das Graphical User Interface (GUI) und Komponenten für die Sprachsteuerung unterschieden. Da die Sprachkomponenten jedoch für diese Arbeit nicht weiter relevant sind werden diese in den folgenden Kapiteln auch nicht genauer erläutert. Innerhalb des GUI Bereiches bildet EB GUIDE Studio das tatsächliche Modellierungstool mit dem das Verhalten und Aussehen der Benutzeroberfläche definiert wird. Für das entwickelte Modell stellt das EB GUIDE Target Framework auf dem Zielsystem die Laufzeitumgebung bereit.[[Elek b](#)]

EB Guide Studio ist das Interface von EB Guide mit dem nach dem What-You-See-Is-What-You-Get (WYSIWYG) Prinzip User Interfaces modelliert werden. Durch das WYSIWYG

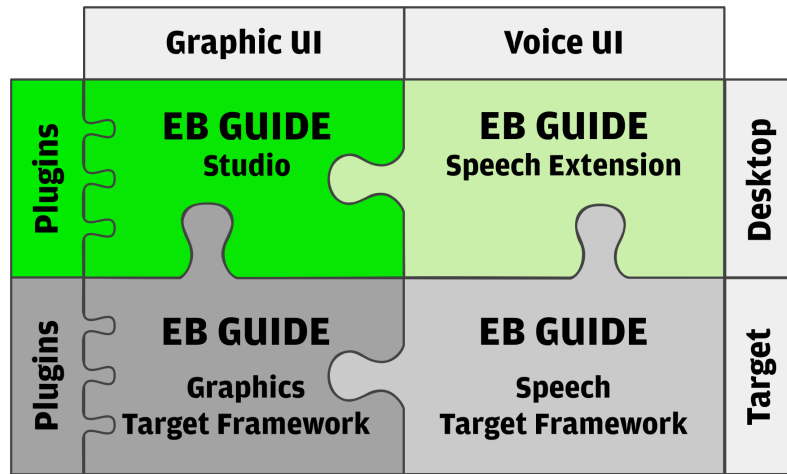


Abbildung 2.3.: Aufbau EB GUIDE

Prinzip ist es während des Modellierens einer View bereits möglich das Endergebnis des Designs zu sehen. Das Verhalten des Interfaces hingegen wird mithilfe einer Zustandsmaschine, der sogenannten Statemachine, modelliert die auf dem UML- Prinzip aufbaut. Die Trennung der Logik und des Designs wird in EB GUIDE Studio grafisch durch zwei unterschiedliche Arbeitsoberflächen gestaltet in welchen den Modellierern jeweils die entsprechenden Tools und Elemente zur Verfügung stehen.

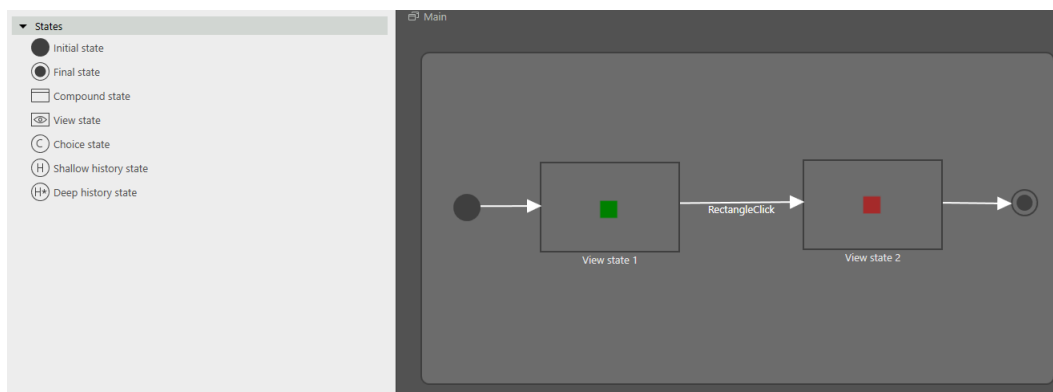


Abbildung 2.4.: EB Guide Studio Statemachine

In Abb. 2.4 ist die Statemachine für ein simples Beispiel zu sehen. Links neben der Arbeitsfläche befindet sich eine Toolbox mit deren Inhalt per Drag and Drop auf der Arbeitsfläche die benötigte Logik definiert wird. Wie bei UML-Diagrammen gibt es einen Initial State der

den Startpunkt angibt und einen Final State der die Statemachine beendet. Die ebenfalls zu sehenden View States stehen für einen Screen im Endprodukt, dementsprechend wird in den View States auch das Aussehen der Interfaces definiert. Die Verbindungen zwischen den States werden als Transitionen bezeichnet und mithilfe von Events ausgelöst. Events stellen hierbei beliebige Ereignisse dar die durch Elemente in der View ausgelöst werden können. Der Auslöser für das Event wird mithilfe einer eigens für EB Guide entwickelten Skriptsprache als Trigger für dieses gesetzt. Die häufigsten Auslöser sind Benutzeraktionen mit Widgets, die in Abb. 2.5 zu sehen sind. Widgets sind Elemente mit denen das Aussehen des Interfaces im sogenannten View Editor bestimmt wird und die sich in Basis- und 3D-Widgets einteilen lassen. Alle Basiswidgets verfügen über Basiseigenschaften wie Höhe, Breite und Farbe sowie über spezifische Eigenschaften wie zum Beispiel "Touch-Released" bei einem Button. [Elek] Beispielsweise wird das Event "RectangleClick", was sich in Abb. 2.4 an der Transition zwischen View State 1 und View State 2 befindet in diesem Beispiel durch das Klicken auf das grüne Rechteck in View State 1 ausgelöst. Ist die Benutzeraktion abgeschlossen findet der Übergang von View State 1 zu View State 2 statt und der Nutzer sieht nun anstatt dem grünen ein rotes Rechteck. Damit diese Aktion erfolgreich ausgeführt wird muss das grüne Rechteck die Eigenschaft "Touch-Released" zugewiesen bekommen und über die EB Guide Skriptsprache mitgeteilt bekommen das Event RectangleClick zu feuern sobald das grüne Rechteck berührt wurde. Da sich dieses Event in der Statemachine an der Transition zwischen den beiden States befindet wird nun durch einen Klick auf das grüne Rechteck ein Bildschirmwechsel zwischen den beiden View States ausgelöst.

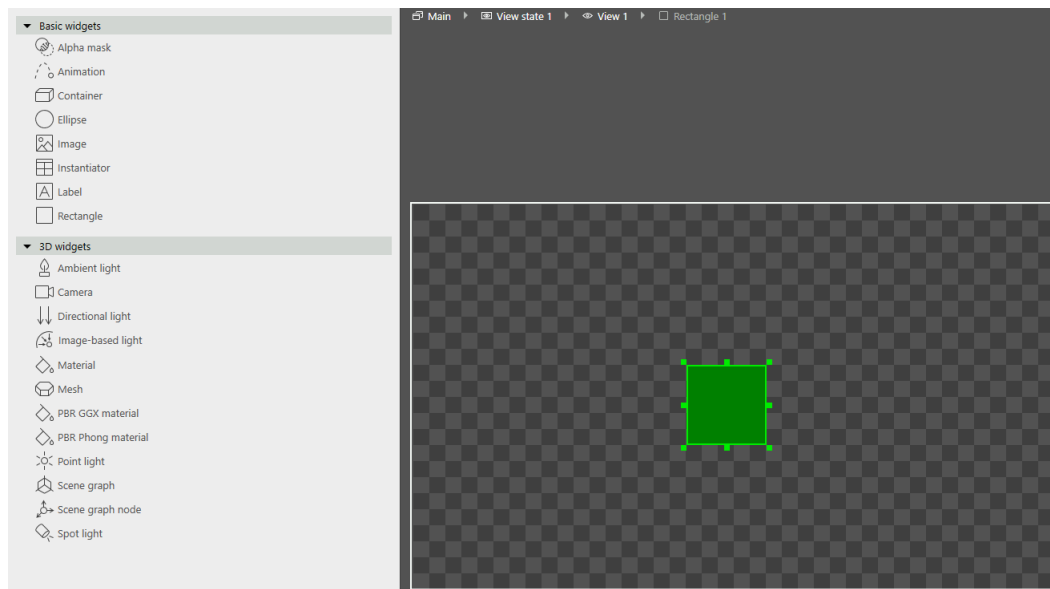


Abbildung 2.5.: EB Guide VIEW

Kapitel 3.

Analysen

3.1. Voruntersuchungen

3.1.1. Ausgangssituation

3.1.2. Vorgehensweise

3.1.3. Ergebnisse

Image List Jedes Listenelement muss einzeln hinzugefügt werden. Listen haben immer große Dimensionen, unglaublicher Zeitaufwand.

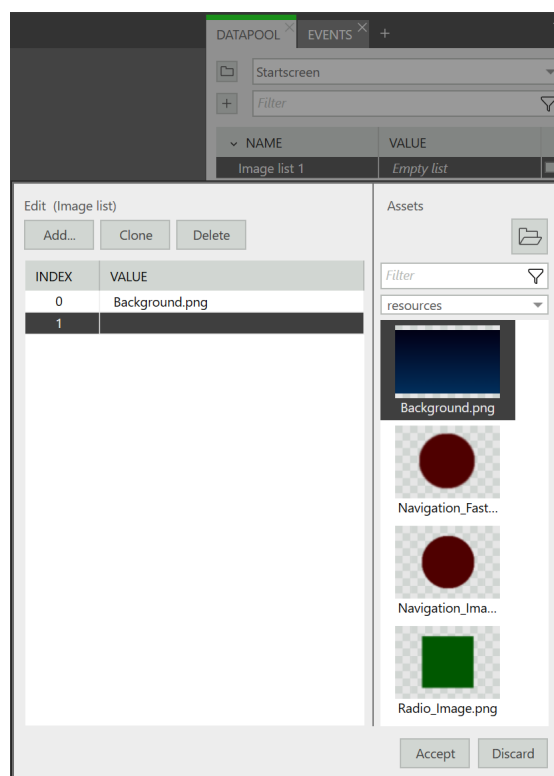


Abbildung 3.1.: Usability - Schwäche Image List

Navigation Wird ein neues Element in der View hinzugefügt wird der Baum in der Navigation nicht automatisch ausgeklappt. Das Element muss zum umbenennen gesucht werden. Automatisches ausklappen und markieren des eingefügten Objektes zur Umbenennung.

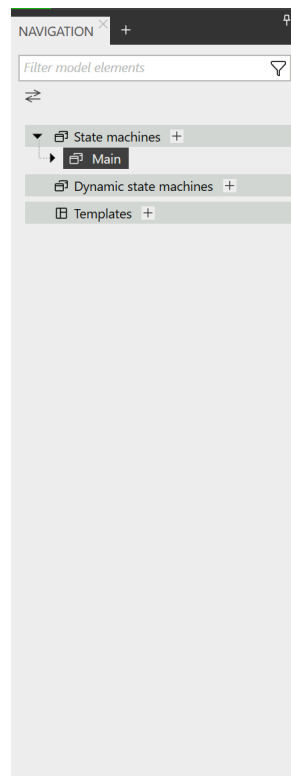


Abbildung 3.2.: Usability - Schwäche Navigation

Template Properties Rechtsklick bei Widget Properties notwendig um Verlinkung zum Template Interface zu erstellen. Einfacher klick auf Kreis vorne?

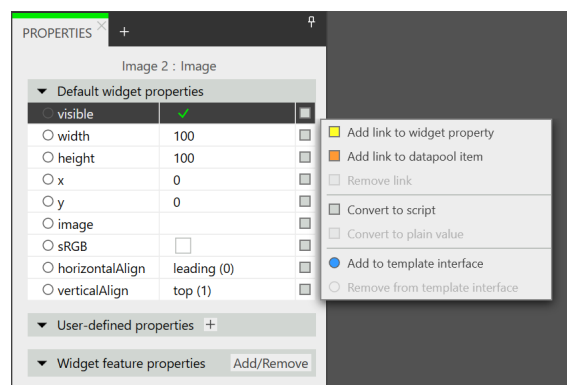


Abbildung 3.3.: Usability - Schwäche Template Properties

Widget Feature Properties Häufiges Ausklappen der obermenüs bei der Suche nach bereits bekannten Features. Suche einbauen?

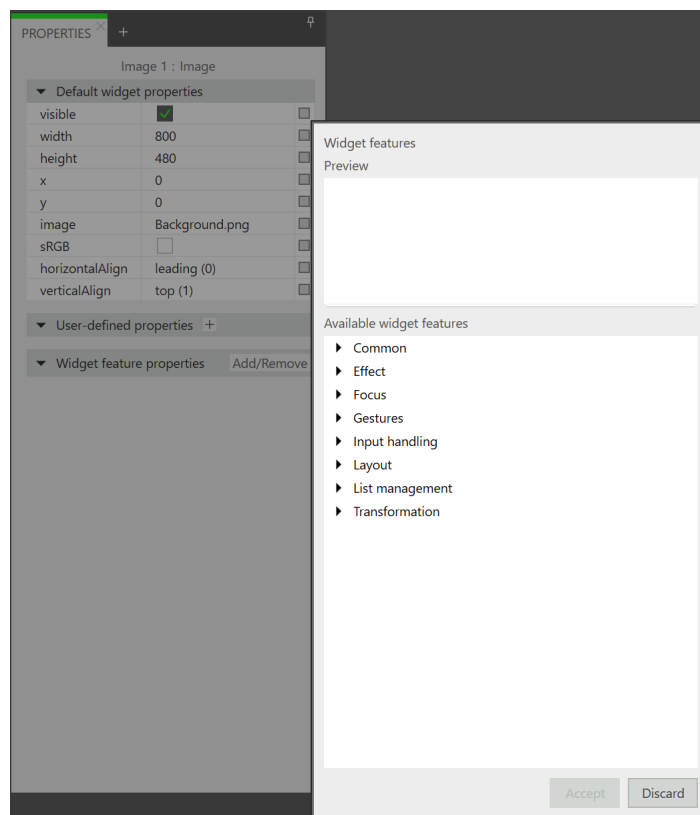


Abbildung 3.4.: Usability - Schwäche Widget Feature Properties

Mehrfachselektion Bei Mehrfachselektion keine Anpassung der Properties mehr möglich. Im WidgetTree nicht sichtbar was ausgewählt ist, hier ist eine mehrfache Markierung auch nicht möglich.

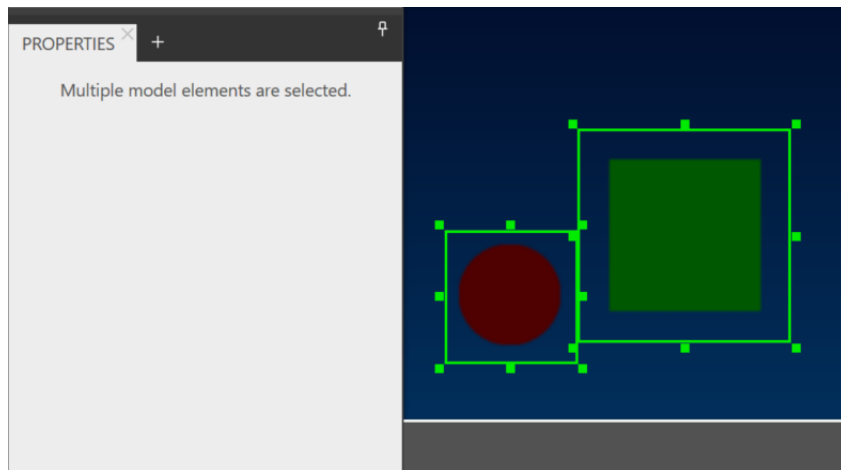


Abbildung 3.5.: Usability - Schwäche Mehrfachselektion

Default Widget Properties Springen zum nächsten Property mit Enter/Pfeiltasten möglich, jedoch keine Markierung zur direkten Bearbeitung. Auch hier keine Mehrfachselektion möglich, praktisch bei z.B. quadratischen Bildern.

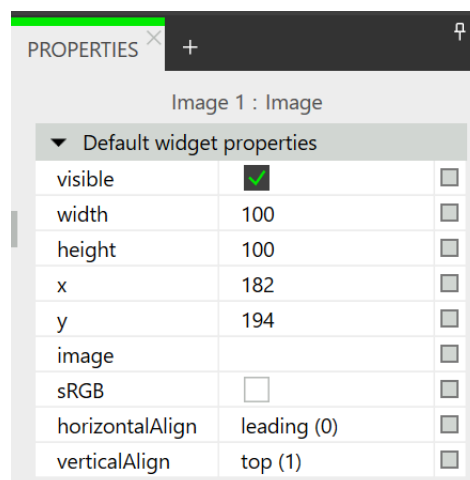


Abbildung 3.6.: Usability - Schwächen bei den Default Widget Properties

3.2. Verbesserungen

3.2.1. Auswahlkriterien

3.2.2. Gewinn für den Nutzer

3.2.3. Design der Verbesserungen

Navigation

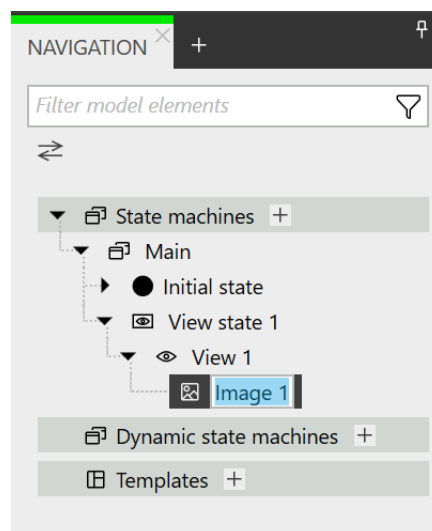


Abbildung 3.7.: Verbesserung Navigation

Template Properties

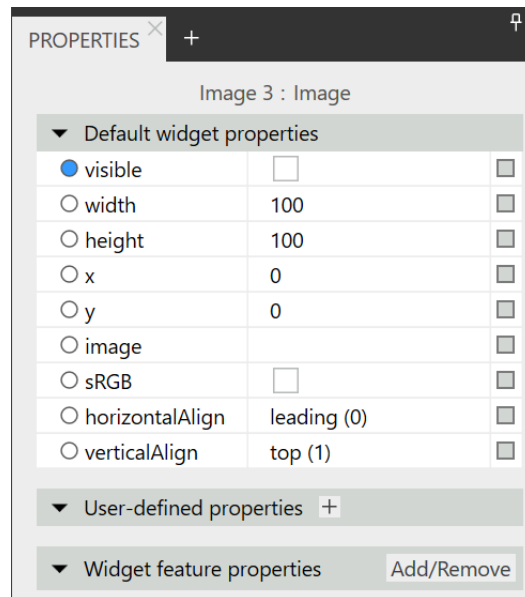


Abbildung 3.8.: Verbesserung Template Properties

Widget Feature Properties

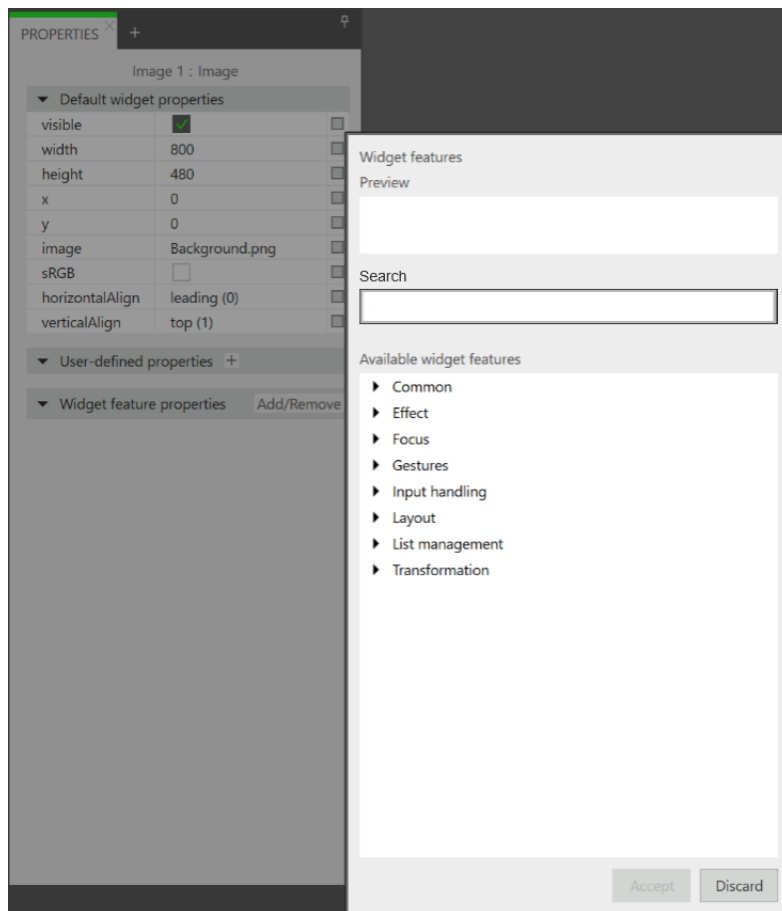


Abbildung 3.9.: Verbesserung Feature Property Properties

Mehrfachselektion

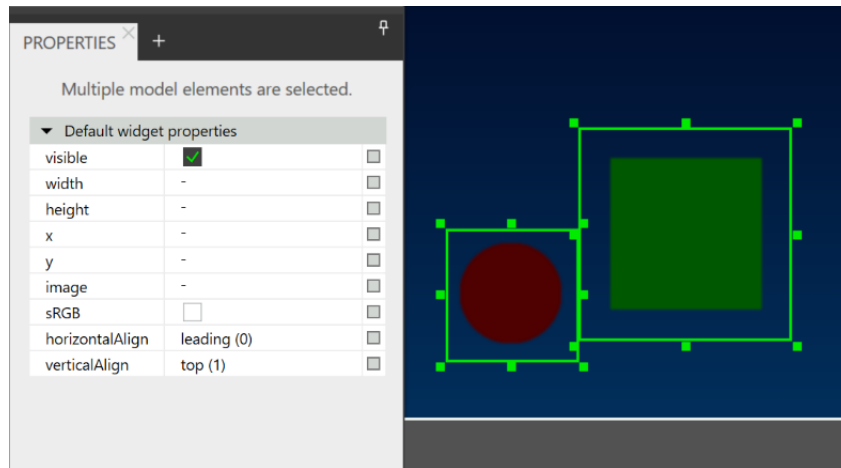


Abbildung 3.10.: Verbesserung Mehrfachselektion

Default Widget Properties

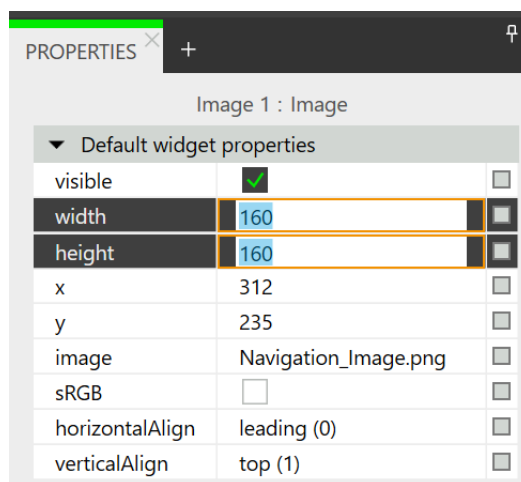


Abbildung 3.11.: Verbesserung Default Widget Properties

Kapitel 4.

Umsetzung

4.1. Prototyp Multiselektion

4.1.1. Zielsetzung

4.1.2. Axure RP

4.1.3. Interaktionsmöglichkeiten

4.1.4. Vorgehensweise

4.2. Implementierung Filter

4.2.1. Zielsetzung

4.2.2. Projektaufbau

4.2.3. Vorgehensweise

Kapitel 5.

Usability - Test

5.1. Lookback

5.2. Remote Usability - Test

5.3. Arbeitsaufgaben

5.4. Ergebnisse altes Interface

5.5. Ergebnisse überarbeitetes Interface

5.6. Vergleich

Kapitel 6.

Fazit

6.1. Ergebnisse

6.2. Ausblick

Anhang A.

Supplemental Information

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Abbildungsverzeichnis

2.1.	Zusammenhang Usability und User Experience (nach [Saro 16])	5
2.2.	Human - Centered - Design - Process bei Elektrobot	5
2.3.	Aufbau EB GUIDE	11
2.4.	EB Guide Studio Statemachine	11
2.5.	EB Guide VIEW	12
3.1.	Usability - Schwäche Image List	13
3.2.	Usability - Schwäche Navigation	14
3.3.	Usability - Schwäche Template Properties	14
3.4.	Usability - Schwäche Widget Feature Properties	15
3.5.	Usability - Schwäche Mehrfachselektion	16
3.6.	Usability - Schwächen bei den Default Widget Properties	16
3.7.	Verbesserung Navigation	17
3.8.	Verbesserung Template Properties	18
3.9.	Verbesserung Feature Property Properties	18
3.10.	Verbesserung Mehrfachselektion	19
3.11.	Verbesserung Default Widget Properties	19

Tabellenverzeichnis

List of Listings

Literaturverzeichnis

- [Eleka] Elektrobit Automotive GmbH. “About Elektrobit (EB) - Elektrobit”.
- [Elekb] Elektrobit Automotive GmbH. “EB GUIDE Home”.
- [Elekc] Elektrobit Automotive GmbH. “EB GUIDE Studio - User guide”.
- [Ergo 14] “Ergonomie - was ist das eigentlich? Definitionen und Forschungsrichtungen”. 2014.
- [Knig 19] W. Knight. *UX for Developers*. Apress, Berkeley, CA, 2019.
- [Lidw 10] W. Lidwell, K. Holden, and J. Butler. *Universal principles of design: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design / William Lidwell, Kritina Holden, Jill Butler*. Rockport, Beverly, Mass., [rev. and updated] ed. Ed., 2010.
- [Niel95] J. Nielsen. *Usability engineering*. AP Professional, Boston and London, [new ed.] Ed., 1995?
- [Norm 16] D. A. Norman. *The design of everyday things*. Verlag Franz Vahlen, München, überarbeitete und erweiterte auflage Ed., 2016.
- [Rich 16] M. Richter. *Usability und UX kompakt [electronic resource]: Produkte für Menschen. IT kompakt*, Springer Vieweg, Berlin, 4. auflage Ed., 2016.
- [Saro 16] F. Sarodnick and H. Brau. *Methoden der Usability Evalution: Wissenschaftliche Grundlagen und praktische Anwendung*. Hogrefe, Bern, 3., unveränderte auflage Ed., 2016.