



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Entwicklung einer Benutzungsoberfläche zur Erstellung von Oberflächenentwürfen oder Web Site Gestaltung mittels eines Grafiktablets oder Tablet PC

Studienarbeit

im Studiengang Computervisualistik

vorgelegt von
Daniela Holl

Betreuer: Dipl. Inf. Holger Heuser
Institut für Computervisualistik

Koblenz, im Juli 2007

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....
(Ort, Datum) (Unterschrift)

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 3 |
| 2 | State of the Art | 5 |
| 2.1 | Microsoft Office Visio Professional 2003 | 5 |
| 2.2 | SmartDraw | 9 |
| 2.3 | GuiGenie | 13 |
| 2.4 | Mockup Screens | 16 |
| 2.5 | Denim | 20 |
| 2.6 | Vergleich | 23 |
| 3 | Entwicklung der Benutzungsoberfläche | 27 |
| 3.1 | Gestaltung und Aufbau der Benutzungsoberfläche | 27 |
| 3.2 | Programmstruktur | 35 |
| 3.2.1 | Die Klassen ScribbleUI und Main | 35 |
| 3.2.2 | Die Klassen Scribblesheet und LineColor | 36 |
| 3.2.3 | Die Klassen ImagePaste, Imagecomponent und Backgroundcomponent | 36 |
| 3.2.4 | Die Klassen ImageFilter, SaveImageFilter und Utils | 37 |
| 3.2.5 | Die Klasse Save | 37 |
| 3.3 | Ausblick | 37 |

Abbildungsverzeichnis

| | | |
|-----|---|----|
| 2.1 | Oberfläche und Komponentenauswahl von MS Visio | 8 |
| 2.2 | Beispiel eines mit SmartDraw erstellten Oberflächenentwurfs | 12 |
| 2.3 | Übersicht über die verfügbaren Komponenten | 15 |
| 2.4 | Verfügbare Elemente im Screen Editor | 18 |
| 2.5 | Übersicht über den Aufbau und die verfügbaren Werkzeuge | 23 |
| 3.1 | Übersicht über die Funktionen der Karteikarte Datei | 29 |
| 3.2 | Dialog zur Auswahl eines Hintergrundes | 30 |
| 3.3 | Übersicht über die Funktionen der Karteikarte Bearbeiten | 30 |
| 3.4 | Inhalt der Kategorie Zeichnen | 32 |
| 3.5 | Inhalt der Kategorie Textfeld | 32 |
| 3.6 | Zusätzliche Einstellungen für das Stiftwerkzeug | 32 |
| 3.7 | Zusätzliche Einstellungen für das Textwerkzeug | 33 |
| 3.8 | Darstellung des Zoombereichs | 34 |
| 3.9 | Oberfläche von Scribble It | 34 |

Kapitel 1

Einleitung

Im Rahmen dieser Studienarbeit wird eine graphische Benutzungsoberfläche mit grundlegenden Funktionalitäten entwickelt. Diese soll es einem Benutzer ermöglichen, einfache Oberflächenentwürfe zu erstellen mit Hilfe eines Grafiktablets oder eines Tablet PC als Ersatz für Stift und Papier. Hierzu wird zunächst ein State of the Art Bericht über die Softwareprodukte Denim, MockupScreens, MS Visio, SmartDraw und GuiGenie erstellt werden. In diesem Bericht werden die Tools unter softwareergonomischen Gesichtspunkten betrachtet und die Nützlichkeit ihrer Komponenten bewertet.

Aufbauend auf den daraus gewonnenen Erkenntnissen, in Anlehnung an das Denim Projekt der Universität Washington¹ und unter Berücksichtigung des WOB-Modells, wird eine Benutzungsoberfläche gestaltet und entwickelt. Weiterhin werden grundlegende Funktionalitäten der erstellten Oberfläche, wie zum Beispiel das Speichern erstellter Dateien oder das Laden von Bildern und Verwenden von Bildern als Hintergrund, implementiert. Zudem soll ein Werkzeug zum Zeichnen von Freihandlinien zur Gestaltung einer Oberfläche, beispielsweise einer Web Site, entwickelt werden.

Im Anschluß an die Gestaltungs- und Entwicklungsphase werden die Gestaltung und die Funktionen der Benutzungsoberfläche in schriftlicher Form festgehalten.

¹ehemals Berkeley

Kapitel 2

State of the Art

Im State of the Art Bericht werden zunächst fünf Softwareprodukte zur Visualisierung von Oberflächen in einem jeweils eigenen Abschnitt vorgestellt. Dies sind aus dem kommerziellen Bereich SmartDraw, MS Visio und MockupScreens, sowie aus dem nicht kommerziellen Bereich die Produkte Denim und GuiGenie.

Abschließend werden die fünf Produkte einander gegenübergestellt und hinsichtlich ihrer Bedienbarkeit, ihrer Selbsterklärungsfähigkeit, ihres Umfangs, sowie ihrer Kosten verglichen und bewertet.

2.1 Microsoft Office Visio Professional 2003

MS Office Visio Professional 2003 ist ein kommerzielles Produkt der Firma Microsoft. Es bietet dem Benutzer eine sehr große Auswahl graphischer Visualisierungsmöglichkeiten und deckt dadurch ein breites Spektrum einer Vielzahl möglicher Anwendungsgebiete ab. Dies sind beispielsweise Schaltkreise im Bereich der Elektrotechnik, Grundrisse im Bereich der Bauplanung, die Modellierung von Datenbanken und Geschäftsprozessen, oder die Erstellung dreidimensionaler Wegbeschreibungen, um nur einige Anwendungsgebiete, außer dem des Softwaredesigns zu nennen. Im Folgenden werden die Unterkategorie Windows XP-Benutzeroberfläche der Kategorie Software, sowie die Unterkategorien konzeptionelle Website und Websiteübersicht, der Kategorie Webdiagramm, vorgestellt.

Bei Programmstart besteht die Oberfläche aus dem Menü, einer Toolbar, sowie drei Spalten. In der linken Spalte sind die Kategorien aufgelistet, zu denen sich in der mittleren Spalte jeweils Vorlagenbeispiele mit den jeweiligen Unterkategorien befinden. In der rechten Spalte befindet sich die Hilfe. Nachdem eine Unterkategorie gewählt wurde, ändert

sich die Oberflächenaufteilung, so dass sich in der linken Spalte die Auswahlfelder mit den Komponenten befindet, die jeweils durch einen Button, der mit dem jeweiligen Oberbegriff der Komponentenauswahl bezeichnet ist, repräsentiert werden und in der mittleren Spalte die von einem Lineal umrandete Arbeitsfläche. Das Menü, die Toolbar und die Hilfe bleiben in ihrer Anordnung unverändert.

Das Menü besteht aus neun Menüpunkten und besitzt bis zu drei Hierarchieebenen, beispielsweise beim Menüpunkt *Datei* und seinem Eintrag *Neu*, da hier erneut die Möglichkeit zur Wahl einer Unterkategorie besteht. In diesem Menüpunkt befindet sich zudem der Eintrag *Shapes*, mit welchem sich der Benutzer die unterschiedlichen Komponentenkategorien in die Auswahl laden kann. Da es jedoch einen eigenen Menüpunkt *Shapes* gibt, ist es fraglich, warum dieser Eintrag in *Datei* realisiert wurde und nicht im Menüpunkt *Shape*. Die Anzahl der einzelnen Einträge in der ersten Ebene variieren von fünf bis zu siebzehn Einträgen, wovon einige Einträge, beispielsweise der Eintrag *Symbolleiste* unter dem Menüpunkt *Ansicht*, selbst auf eine Ebene mit siebzehn Einträgen verweist. Das Menü ist dementsprechend sehr unübersichtlich und in Kombination mit der Toolbar redundant, da in dieser ein Teil der Menüeinträge bereits realisiert ist. Die Toolbar kann jedoch benutzerdefiniert angepasst werden, so dass ein Benutzer, der bereits Erfahrung mit Visio hat, diese seinen Bedürfnissen entsprechend ändern kann.

Die Hilfefunktion beinhaltet neben der Hilfedatei eine Suchfunktion, mittels derer der Benutzer gezielt die Hilfedatei durchsuchen kann. Gibt der Benutzer einen Suchbegriff ein, so werden die Suchergebnisse untereinander aufgelistet, sowie weitere Gebiete aus dem gesamten Office Paket, in denen zusätzliche Informationen gefunden werden könnten, aufgelistet. Die weiteren Suchgebiete erhält der Benutzer auch dann, wenn in der Hilfedatei keine Entsprechungen gefunden wurden. Das Suchfenster selbst, welches sich zu Beginn der Eingabe im oberen Bereich der Spalte befunden hatte, erscheint nun unterhalb der Ergebnisliste. Wird die Hilfedatei über ihr Inhaltsverzeichnis verwendet, so muss sich der Benutzer zunächst für ein Kapitel der Hilfe entscheiden. Um die Überschriften der Kapitel vollständig lesen zu können, ist jedoch ein horizontales scrollen notwendig. Da das Inhaltsverzeichnis als Baumstruktur aufgebaut ist, muss nach der Wahl eines Kapitels zusätzlich noch vertikal gescrollt werden, da sich die Spalte für die Anzeige der Hilfe nicht automatisch vergrößert. Dies ist nur manuell möglich, bis zu einem Maximum der halben Arbeitsfläche. Zudem muss sich der Benutzer innerhalb eines Kapitels der Hilfedatei teilweise noch durch die Unterkapitel, deren Abschnitte und Unterabschnitte, durchklicken, bis er die mit dem *Hilfe-Ikon* markierten Einträge, welche durch eine blaue Schriftfarbe gegenüber den in schwarzer Schrift gehaltenen Einträgen, zusätzlich hervorgehoben

werden, findet. Wird ein so markierter Eintrag angewählt, öffnet sich ein neues Fenster, in welchem der Hilfetext enthalten ist. Für die Navigation stehen dem Benutzer ein *Vor*- und ein *Zurückbutton* zur Verfügung, sowie ein Button *Homepage*, um zur Online-Hilfe zu gelangen. Die Hilfetexte selbst sind kurz gehalten und beinhalten zwar weiterführende Links zu dem gewählten Thema, jedoch keine graphischen Beispiele. Die Hilfefunktion kann, um beispielsweise die Arbeitsfläche zu vergrößern, ausgeblendet werden. Desweiteren öffnet sich zu jeder Komponente in den Auswahlfeldern, jedem Button oder Ikon ein Popup-Fenster mit der jeweiligen Bezeichnung. Bei den Komponenten wird zusätzlich noch eine kurze Hilfetext angezeigt.

Die linke Spalte der Oberfläche, *Shapes*, beinhaltet neben einer Suchfunktion für die verschiedenen *Shapes* selbst, die bereits erwähnten Buttons, welche jeweils ein Auswahlfeld enthalten. Der Benutzer kann diesen Bereich ebenfalls an seine Bedürfnisse anpassen, indem er einzelne Auswahlfelder schließt, oder über den Eintrag *Shapes* unter dem Menüpunkt *Datei*, weitere Auswahlfelder in den Bereich lädt. Bei Auswahl eines Buttons öffnet sich die zugehörige Komponentenauswahl direkt unterhalb des Buttons. Die Komponenten sind sowohl graphisch dargestellt, als auch benannt. Der Benutzer kann die Komponenten sowohl einzeln, als auch mehrere Komponenten, unter Verwendung der Shift-Taste, mittels Drag und Drop auf das Zeichenblatt in der Arbeitsfläche ziehen. Die Komponenten sind untereinander frei kombinierbar und können, sofern sie markiert wurden, frei positioniert werden. Bei den Auswahlen *Symbole*, *Websiteübersicht* und *konzeptionelle Website* sind alle Komponenten, bis auf die Pfeile und den Siteübersichtsknoten, welche nur vertikal beziehungsweise nur horizontal skaliert werden können, frei skalierbar. Bei der *Assistentenauswahl* hingegen, können die einzelnen Komponenten nicht skaliert werden, dies gilt ebenso für die Schaltflächen- und Symbolleistentypen der Auswahl *Symbolleisten* und *Menü* gilt. Die Auswahlfelder *Standardsteuerelemente* und *Fenster/Dialogfelder* beinhalten sowohl Komponenten die frei skalierbar, teilweise skalierbar oder gar nicht skalierbar sind. Welche der Komponenten einer Auswahl in ihrer Größe veränderbar sind, kann der Benutzer jedoch nur durch Testen herausfinden. Die Komponenten können teilweise weder mit einem Text versehen werden, noch farblich verändert oder gedreht, beziehungsweise gespiegelt werden. Der Benutzer kann zwar beispielsweise eine neue Farbe für eine Fensterleiste einstellen, diese wird jedoch nicht übernommen, sondern löst eine Warnmeldung aus, in der mitgeteilt wird, dass die Komponente oder die derzeitige Arbeitsebene geschützt ist. Es ist daher für einen Benutzer nicht nachvollziehbar, welche Komponente wie veränderbar ist, da ein Teil der nicht anwendbare Funktionen dennoch ausgewählt werden können.

MS Visio bietet dem Benutzer die Möglichkeit, eigene Shapelemente zu erstellen und diese in einem eigenen Auswahlfeld zu speichern, so dass sie ihm bei späteren Anwendungen als Komponenten zur Verfügung stehen. Hierzu wird ein neues Shapeelement, zum Beispiel ein Button, auf der Arbeitsfläche gezeichnet und dann mittels Drag und Drop in die entsprechende Auswahl gezogen. Besteht die erstellte Komponente aus mehreren Einzelementen, so sollten diese zuvor alle markiert und anschließend gruppiert werden, um es wie eine Komponente verwenden zu können. Die selbst erstellten Komponenten sind frei positionierbar und skalierbar. Zudem können auf diese Komponenten die verfügbaren Funktionen, wie zum Beispiel das Ändern der Füllfarbe, angewendet werden. Zur Erstellung eigener Komponenten können die Standardformen, die Freihandfunktion, das Freihandtool oder eine Kombination mit den vordefinierten Komponenten verwendet werden.

Zur Veranschaulichung der Komponentenauswahl und der Oberflächengestaltung möge die Abbildung 2.1 dienen. Die Auswahl *Schablone2* beinhaltet die selbst erstellten Komponenten.

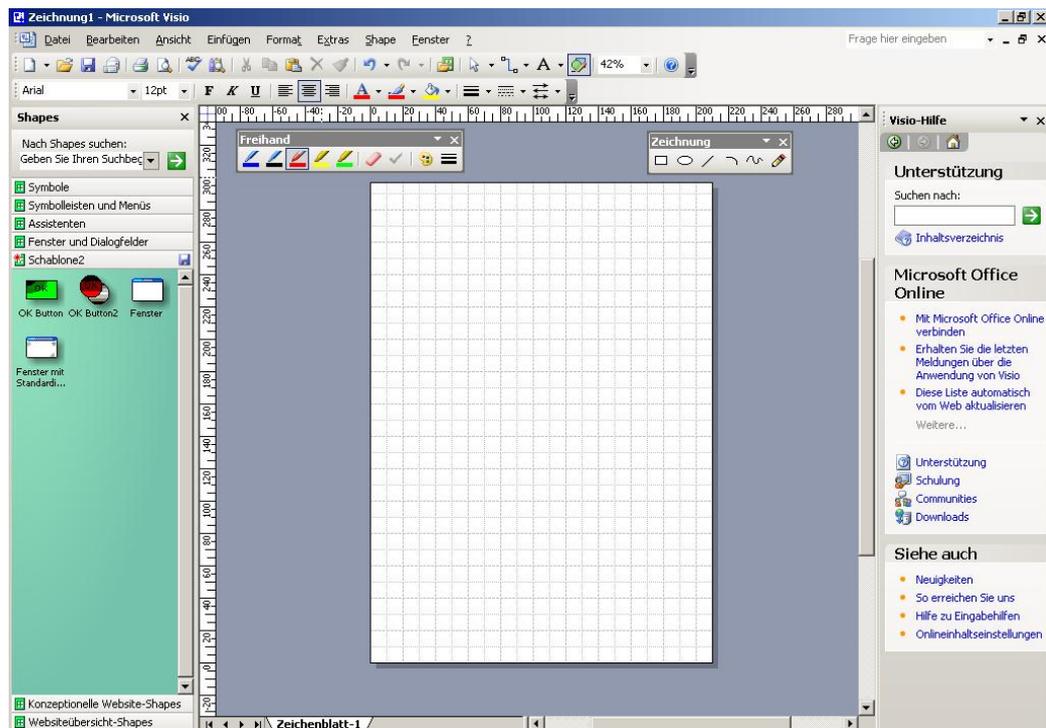


Abbildung 2.1: Oberfläche und Komponentenauswahl von MS Visio

Das *Freihandtool* bietet dem Benutzer die Wahl zwischen einem Kugelschreiber, einem

Textmarker oder einem Filzstift. Zudem kann der Benutzer die Linienfarbe und Strichstärke einstellen. Eine spätere Änderung der Linienfarbe und der Strichstärke von bereits gezeichneten Objekten ist jedoch nicht möglich. Irreführend bei der *Freihandfunktion* ist, dass sich im Tool *Zeichnen*, ein *Freihandzeichnen* befindet und es zusätzlich noch das Tool *Freihand* gibt, welches die Stifte beinhaltet. Beim *Zeichnentool*, glättet das *Freihandzeichnen* die gezeichneten Objekte, so dass beim zeichnen eines Rechtecks, die Ecken stets abgerundet sind. Das *Freihandzeichnen* des *Zeichnentools* ist daher eher für geschwungene Formen geeignet. In der *Zeichnenfunktion* stehen dem Benutzer außer den bereits erwähnten Standardformen und der *Freihandfunktion*, noch die Funktionen *Linien*, *Bogen* und *Bleistift*, der eine Kombination von *Linien* und *Bogen* ist, zur Verfügung. MS Visio bietet dem Benutzer eine große Vielfalt graphischer Anwendungsmöglichkeiten der unterschiedlichsten Visualisierungsgebiete, erfordert aber, um diese in vollem Umfang nutzen zu können eine längere Einarbeitungszeit. Bei einer ausschließlichen Verwendung der vorgegebenen Shapeelemente wird der Benutzer bei der Gestaltung einer graphischen Benutzeroberfläche jedoch auf das Windows XP Design eingeschränkt. Ein individuelles Design ist daher nur möglich, wenn der Benutzer die entsprechenden Komponenten eigenhändig erstellt.

2.2 SmartDraw

SmartDraw 2007 ist ein kommerzielles Produkt der Firma SmartDraw.com, welches für viele verschiedene Gestaltungsbereiche konzipiert wurde. So bietet es neben der Gestaltung von Softwaredesigns, unter anderem die Möglichkeit, Stammbäume oder Landschaften zu gestalten. Im weiteren Verlauf werden nur die Unterkategorien GUI und Web Design der Kategorie Software Design betrachtet.

Die Oberfläche des Programms ist in drei Spalten aufgeteilt. Die rechte Spalte beinhaltet die Hilfefunktion, den Hilfetext, sowie zwei Navigationsbuttons zum Vor- und Zurückblättern innerhalb der Hilfe. Der Hilfetext kann geschlossen werden, um mehr Platz in der mittleren Spalte, dem späteren Arbeitsbereich, zu erhalten. Er kann durch drücken des *SmartHelp-Buttons* jederzeit wieder eingeblendet werden. Der Hilfetext wird zudem auf die Wahl der Kategorie und der jeweiligen Subkategorie angepasst, so dass der Benutzer nicht die seiner Wahl entsprechende in der Hilfe suchen muss.

Die linke Spalte beinhaltet die Gestaltungskategorien, ein Suchfeld, sowie einen Button zur Anzeige der Suchergebnisse. Nachdem der Benutzer eine Kategorie gewählt hat, erscheinen ihre Subkategorien darunter aufgelistet. In der mittleren Spalte befinden sich alle

vorhandenen Beispiele und Vorlagen zu den Subkategorien, die mittels vertikalem scrollen betrachtet und durch anklicken ausgewählt werden können. Zu jedem der Beispiele erscheint eine Quickinfo, wenn der Cursor über dem Beispiel verweilt.

Der Benutzer hat zwei Möglichkeiten eine Subkategorie auszuwählen. Einerseits kann er in der linken Spalte die gewünschte, durch anklicken, auswählen und sich dann eine Vorlage aus der mittleren Spalte aussuchen. Andererseits, kann er direkt in der mittleren Spalte eine Vorlage durch anklicken auswählen. Sollte der Benutzer mit seiner Auswahl nicht zufrieden sein, kann er dies durch die Betätigung des *Back-Buttons*, welcher sich am linken oberen Bildschirmrand befindet, rückgängig machen und eine erneute Auswahl treffen. Desweiteren befindet sich im oberen Rand der Oberfläche ein *Vor-Button*, der Titel des Projektes, sowie Buttons zum Schließen, Verkleinern und Minimieren der Oberfläche. Wurde eine Vorlage ausgewählt, erscheinen am oberen Rand der Oberfläche fünf Gruppen von Buttons: Software Design, Document, Edit, Arrange und Design. Die Gruppe Software Design beinhaltet sechs Buttons: *Add Symbols*, *Add Lines*, *Add Text*, *Arrow Heads*, *Line Hops* und *SmartHelp*. Jeder dieser Buttons wird verbal beschrieben und durch ein Icon dargestellt. Bewegt man den Cursor über die Buttons, wird jeweils der betreffende grün hinterlegt. Zusätzlich verfügen die Buttons über die bereits erwähnte Quickinfo. Gleiches gilt für die anderen vier Buttongruppen. Bei den Gruppen *Document*, *Edit* und *Arrange* befindet sich jeweils ein Dropdown Menü, welches zusätzliche Funktionen zu den Buttons bietet. Die Gruppe *Design* hingegen wurde als Karteireiter aufgebaut. *Design* selbst enthält, neben den Buttons, noch eine Auswahl an verschiedenen *Styles* für ein gewähltes Objekt. Die Karteikarte *Text* beinhaltet eine Auswahl an verschiedenen Fonts, Schriftgrößen und Textfarben, sechs Ausrichtungsmöglichkeiten, die Einstellung des Zeilenabstands und unterschiedliche Varianten von Aufzählungen. Bei den zwei Letztgenannten und der Textfarbe erscheint wieder eine Auswahl mit Vorlagen. Zudem besitzt dieser Karteireiter ein eigenes Menü, welches zusätzliche Funktionen beinhaltet. Der Karteireiter *Page* beinhaltet sechs Buttons, mit deren Hilfe der Benutzer zum Beispiel ein Gitter über seine Seite legen kann. Auch hier ist wieder ein Menü mit weiteren Funktionen verfügbar. Zudem findet sich hier ein Button zum Suchen und Ersetzen von Wörtern, den ein Benutzer meines Erachtens im Karteireiter *Text* erwarten würde. Der Karteireiter *Insert*, besitzt dagegen kein zusätzliches Menü oder Auswahlfenster sonder lediglich sechs Buttons mit denen der Benutzer beispielsweise eine Tabelle oder ein Bild einfügen kann. Der Reiter *Export* ermöglicht es dem Benutzer seine Seite, oder einzelne Objekte davon, in andere Formate zu exportieren, zum Beispiel in Adobe Acrobat PDF, in ein Worddokument, als JPEG, in HTML oder PowerPoint. Eine kleine Auswahl an Formaten wurden als Buttons

realisiert und die weiteren zur Verfügung stehenden Formate innerhalb eines Menüs. Der Reiter *Table* beinhaltet sechs Buttons und ein Menü zur Erstellung und Bearbeitung von Tabellen. Im letzten Karteireiter, *Help*, befinden sich Buttons für den technischen Support, Updates, User Guide, About, Licence und Buy.

Die linke Spalte der Oberfläche beinhaltet wieder das Suchfeld, einen Button *More Symbols*, sowie zwei Buttons *Webpage Components* und *Web Layout*. Wählt man einen der beiden Buttons, zum Beispiel *Webpage Components*, erscheint unter diesem eine Auswahl der verfügbaren Komponenten, welche mittels Drag und Drop auf die Vorlage gezogen werden können. Jede der Komponenten ist graphisch dargestellt und benannt. Zudem erscheint auch bei ihnen wieder ein Popup Fenster mit einer kurzen Erläuterung. Wählt man nun den Button *Web Layout*, erscheint die zugehörige Auswahl und ersetzt die vorherige. Jede Komponente, die entweder durch Drag und Drop oder bereits in der Vorlage enthalten ist, kann vom Benutzer verändert werden. Sie können durch markieren der Komponente in der Vorlage verschoben, skaliert oder gelöscht werden.

Der Oberflächenaufbau und seine bisher beschriebenen Komponenten, sind für die Unterkategorien *GUI* und *Web Design* identisch. Unterschiede zwischen diesen beiden treten erst nach betätigen des Buttons *More Symbols* auf, da aus dem erscheinenden Menü zusätzliche Komponentenlisten für *GUI* beziehungsweise *Web Design* geladen werden können. Anzumerken ist jedoch, dass sowohl die spezifischen Listen für *GUI* als auch die für *Web Design*, frei kombinierbar sind. Dies bedeutet, dass alle verfügbaren Listen für diese beiden Unterkategorien in der linken Spalte als Buttons angelegt werden können. Nachteilig daran ist, dass dadurch der Platz zur Anzeige der Auswahl deutlich verkleinert wird und der Benutzer vertikal scrollen muss, um alle Komponenten einer Auswahl sehen zu können.

Ein Benutzer kann sich nur unter Verwendung der Einzelkomponenten sehr schnell einen professionell wirkenden Entwurf für eine GUI beziehungsweise eine Website mittels Drag und Drop zusammenstellen. Als Beispiel dient hierzu die Abbildung 2.2, in welcher der erstellte Oberflächenentwurf rot umrandet wurde.

Nachteilig bei dieser Vorgehensweise ist jedoch, dass sich die *GUI* spezifischen Komponenten nicht farblich verändern lassen und nur sehr wenige der *Web Design* Komponenten, welche man nur durch Testen herausfinden kann. Es ist zwar möglich, allen Komponenten beispielsweise einen andersfarbigen Schatten oder einen Text hinzuzufügen, aber ihre Füllfarben sind zumeist unveränderlich. Dies kann jedoch gelöst werden, indem der Benutzer die von ihm gewünschten Komponenten selbst erstellt. SmartDraw bietet hierzu einige Standardformen an, welche jeweils beliebig veränderbar sind. Beispielsweise kann

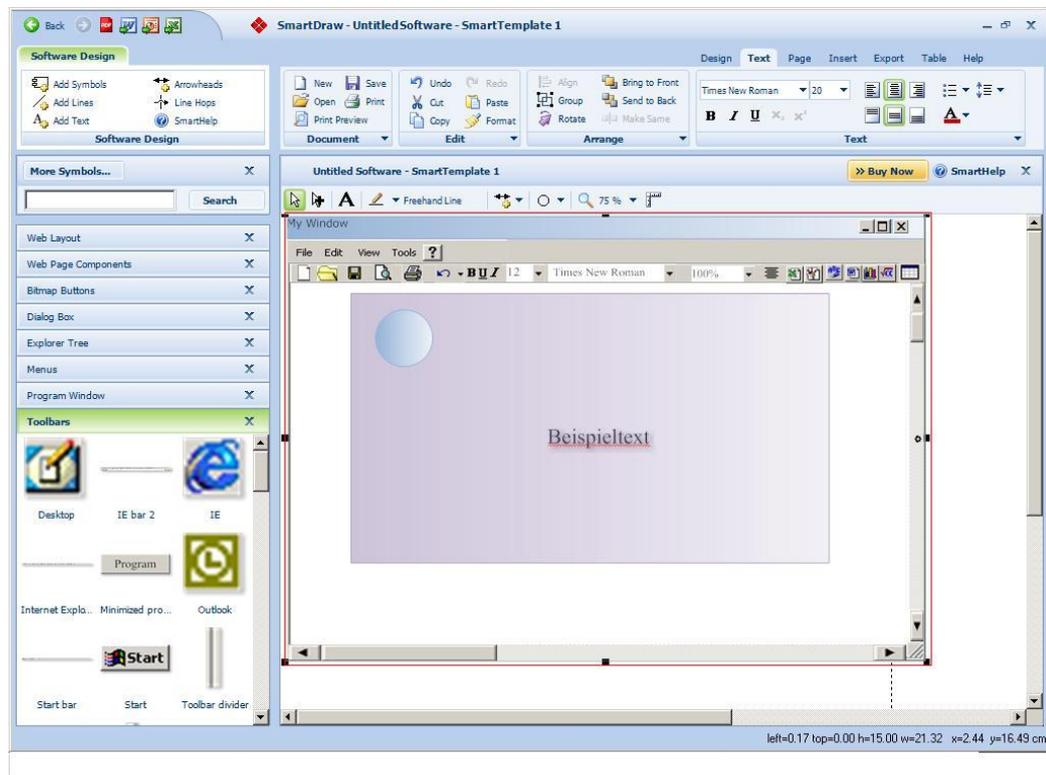


Abbildung 2.2: Beispiel eines mit SmartDraw erstellten Oberflächenentwurfs

man ihnen einen Farbverlauf, Schatten, Reflektionen, Glanzpunkte oder einen glühenden Rand zuweisen. Die Standardformen können zudem auch vollständig in der Auswahl in der linken Spalte angezeigt werden, so dass ein Benutzer auch diese mittels Drag und Drop auf die Arbeitsfläche ziehen kann. Um einer erstellten Oberfläche ein einheitliches Design zu verleihen, muss der Benutzer nicht jede Komponente einzeln anpassen, sondern kann hierzu eines der vordefinierten Farbschemata auswählen. Es ist jedoch nicht möglich ein eigenes Farbschema zu erstellen, welches dann auf den gesamten Entwurf angewendet werden kann. Ein Benutzer müsste dann jede Komponente einzeln anpassen. Anzumerken ist, dass die selbsterstellten Komponenten mit den vordefinierten frei kombinierbar sind. Wird jedoch ein Farbschema darauf angewendet, verändern sich zwar die selbsterstellten, bei den *GUI* spezifischen jedoch keine und bei *Web Design* nur sehr wenige der vordefinierten Komponenten.

Eine andere Möglichkeit, eine Oberfläche zu skizzieren, bietet die Verwendung der *Stiftfunktion*, die sich oberhalb des Arbeitsbereiches befindet. Wie bei den Standardformen können alle verfügbaren Funktionen des Stiftes in die Auswahl geladen werden, bis auf

die Funktion des Freihandzeichnens selbst. Exemplarisch soll hier die Umsetzung dieser Funktion vorgestellt werden.

Bei der Verwendung des Freihandzeichnens muss ein Benutzer für jede neu zu zeichnende Komponente, oder bei Absetzen des Stiftes, die Funktion erneut auswählen. Zwar bietet SmartDraw die Möglichkeit, ein so gezeichnetes Objekt über vordefinierte Ankerpunkte zu manipulieren, ein flüssiges Arbeiten ist damit jedoch nicht möglich. Es können zudem keine weiteren Ankerpunkte dem gezeichneten Objekt hinzugefügt werden, so dass die Veränderbarkeit des Objektes in seiner Form stark eingeschränkt ist. Es eignet sich daher meines Erachtens nur für primitive Formen, die in einem Zug gezeichnet werden können, da bei jedem Absetzen, das Tool erneut ausgewählt werden muss. Versucht man beispielsweise einen Buchstaben in Druckschrift damit zu schreiben, müsste man bereits bei einem H, das Tool dreimal auswählen. Für einen skizzenhaften Oberflächenentwurf ist die reine Verwendung der Freihandfunktion daher nicht geeignet. Mit den anderen Funktionen des Stiftetools, beispielsweise der *Polygon Line* ist es zwar möglich, auch komplexere Formen zu zeichnen, diese müssen aber stets eine geschlossene Form ergeben. Zudem können geschwungene Formen, zum Beispiel ein U oder ein Kreis, nur mittels der Freihandfunktion gezeichnet werden.

SmartDraw bietet einem Benutzer eine Vielzahl an unterschiedlichen Gestaltungsmöglichkeiten und deckt somit einen großen Bereich graphischer Visualisierungsgebiete ab. Es bietet jedoch keinen geeigneten Ersatz für die Verwendung von Stift und Papier in der frühen Entwurfsphase einer Oberfläche.

2.3 GuiGenie

GuiGenie ist ein nichtkommerzielles Produkt von Mario Awad, das zum Zeitpunkt der Erstellung des Berichtes in der Version 1.0.0 Beta vorlag und vollständig in Java programmiert wurde. Die Oberfläche von GuiGenie besteht bei Programmstart aus dem Fenstertitel, einem Menü mit den Einträgen *Project*, *Edit*, *Tools*, *Window* und *Help*, einer Toolbar mit drei Icons zum *Öffnen*, *Speichern* und *Neuanlegen von Projekten*, sowie aus zwei Karteireitern *Design* und *Build*, welche im folgenden als Design- beziehungsweise als Buildmodus bezeichnet werden. Die Oberflächenkomponenten sind vertikal zentriert und linksbündig angeordnet.

Der Menüpunkt *Project* enthält, außer den bereits durch die Toolbar abgedeckten Funktionen, eine *Exit* Funktion, um das Programm zu beenden und eine Funktion um das Projekt zu schließen. Beide Funktionen wären durchaus in der Toolbar realisierbar, so dass auf

den Menüpunkt *Project* hätte verzichtet werden können. Bei dem Menüpunkt *Edit* stellt sich ebenso die Frage nach dem Nutzen. Dieser Menüpunkt besitzt nur einen einzigen Eintrag, *Preferences*, mit welchem der Benutzer Pfade für den Explorer, Notepad und den Windows Taschenrechner setzen kann. Versucht man beispielsweise einen anderen Texteditor anzugeben, lässt sich dieser zwar eintragen, bewirkt jedoch keine Änderung. Unter dem Menüpunkt *Tools* kann der Nutzer die Programme Notepad und den Taschenrechner direkt starten, wobei auch hier ein Eintrag in der Toolbar sinnvoller gewesen wäre als ein eigener Menüpunkt. Ebenso verhält es sich mit dem Eintrag *Window*, welcher dazu dient, die Fensteranordnung zu ändern. Der letzte Menüpunkt, *Help*, besitzt ebenfalls nur zwei Einträge. Der eine Eintrag gibt über den Entwickler und die Version der Software Auskunft, der andere beinhaltet nur eine Anleitung für die Erstellung eines Designbeispiels. Um ein neues Projekt anzulegen hat der Benutzer drei Möglichkeiten, durch drücken des Buttons *new Project*, über den Menüeintrag *Project* oder durch Verwendung der Tastenkombination *Strg-N*. Nachdem der Benutzer eine dieser Möglichkeiten angewendet hat, erscheint ein Fenster *New*, in welchem er sich für einen *Layout Manager* oder eine *absolute Positionierung* entscheiden muss. Durch anwählen eines *Radio Buttons* wird ein *Layout Manager* festgelegt. Als Entscheidungshilfe befindet sich unterhalb der *Layout Manager Auswahl* jeweils ein Beispiel zu dem gewählten Layout. Wenn sich der Benutzer für ein Layout entschieden hat, muss er dies durch drücken des *OK-Buttons* bestätigen, oder andernfalls, den *Abbrechen-Button* drücken. In diesem Zusammenhang tritt eine inkonsistente Sprachverwendung auf, da nur die beiden Buttons *OK* und *Abbrechen* in Deutsch, alle anderen Komponenten von GuiGenie jedoch durchgehend in Englisch bezeichnet sind. Dies könnte jedoch darauf zurückzuführen sein, dass die Standardbeschriftung dieser Buttons durch die JRE im verwendeten Language Pack des Betriebssystem Windows ausgelesen wird.

Wurde ein neues Projekt angelegt, erscheinen nun im Designmodus drei nebeneinander angeordnete Fenster. Die Fensteranordnung kann über den Menüpunkt *Window* geändert werden, so dass die drei Fenster auch kaskadierend, oder untereinander angeordnet werden können. Im weiteren Verlauf wird jedoch von der voreingestellten Fensteranordnung ausgegangen. Das linke Fenster, *Layout*, beinhaltet zwei editierbare Felder und zwei Buttons. Die Felder sind mit *HGap* und *VGap* bezeichnet und dienen der Beeinflussung des Abstands zwischen den Komponenten im mittleren Fenster in horizontaler und vertikaler Richtung. Die Buttons sind mit *Add* und *Delete* bezeichnet, wobei der Benutzer eine im mittleren Fenster markierte Komponente mit dem *Delete-Button* löschen kann. Der Button *Add* dient zum hinzufügen von Komponenten im mittleren Fenster. Bei drücken

dieses Buttons erscheint ein Auswahlfenster mit zehn Komponenten, welche graphisch dargestellt und benannt sind. Der Benutzer kann durch anklicken eine Komponente auswählen, welche dann blau unterlegt wird. Anzumerken ist, dass der Button *Add* bei den Layout Managern *Grid Layout* und *Border Layout*, durch einen Button *Replace* ersetzt wird. Dies bedeutet beispielsweise bei Wahl des *Border Layout* für den Benutzer, dass er nicht mehr als fünf Komponenten in seinen Entwurf einbringen kann, da eine Schachtelung der Layout Manager oder der Komponenten nicht möglich ist. Das Fenster für die Auswahl, sowie die zur Verfügung stehenden Komponenten, sind für jedes gewählte Layout identisch. Zur Veranschaulichung der zur Verfügung stehenden Komponenten möge daher die Abbildung 2.3 dienen.

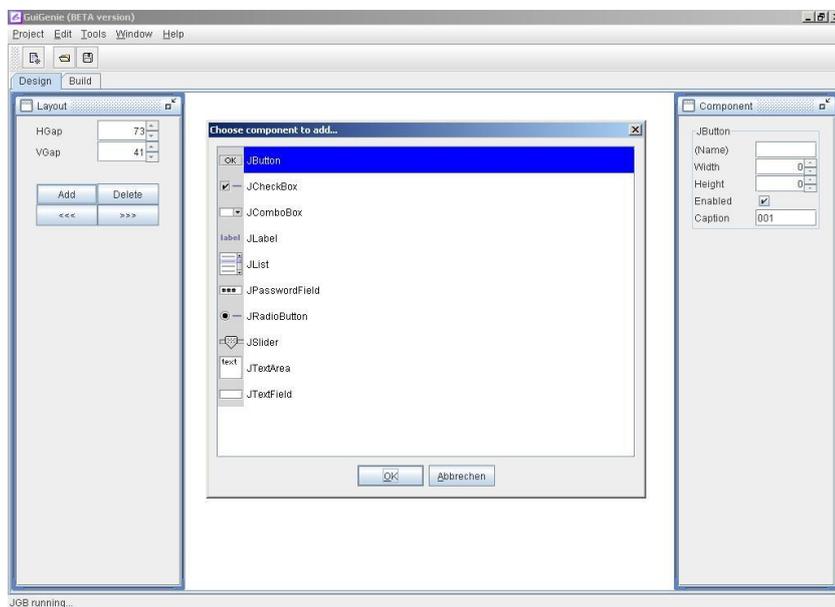


Abbildung 2.3: Übersicht über die verfügbaren Komponenten

Durch drücken des *OK-Buttons* kann der Benutzer die von ihm getroffene Auswahl bestätigen, oder durch drücken des *Abbrechen-Buttons*, verwerfen. Nachdem die gewählte Komponente bestätigt wurde, schließt sich das Fenster und sie erscheint im mittleren Fenster. Nachteilig an dieser Vorgehensweise ist jedoch für den Benutzer, dass er für jede einzelne Komponente die er seinem Oberflächenentwurf hinzufügen möchte, jedesmal erneut den *Add-Button* verwenden muss, um eine weitere Komponente aus dem Fenster auswählen zu können und diese anschließend bestätigen muss. Es ist nicht möglich, mehrere Komponenten in einem Schritt hinzuzufügen oder durch Drag und Drop in das mittlere Fenster zu ziehen. Desweiteren ist ein freies Verschieben der Komponenten nur

dann möglich, wenn der Benutzer anstelle eines Layout Managers die *absolute Positionierung* gewählt hat.

Um eine markierte Komponente in ihrer Höhe und Breite verändern zu können, muss der Benutzer im rechten Fenster, *Components*, die Auswahlfelder *Width* und *Height* entsprechend editieren, entweder durch direkte Eingabe, oder durch verwenden der Pfeiltasten. Im rechten Fenster befinden sich zudem die Textfelder *Name* und *Caption*, in welche der Benutzer eine beliebige Folge von Ziffern und Buchstaben eingeben kann. Das Textfeld *Caption* dient der Beschriftung der Komponenten. Die Eingabe erscheint im Designmodus direkt im mittleren Fenster auf der entsprechenden Komponente. Eine Eingabe im Textfeld *Name* bewirkt in diesem Modus jedoch keine sichtbare Änderung. Diese wird erst im Buildmodus, nach Anwählen des Buttons *Preview Code* und betrachten des generierten Quellcodes, ersichtlich. Im Buildmodus kann der generierte Code angezeigt, gespeichert und kopiert werden. Es ist einem Benutzer jedoch nicht möglich, diesen Code in GuiGenie zu bearbeiten.

GuiGenie eignet sich meines Erachtens mehr für den Entwurf spezifischer Komponenten einer Oberfläche, beispielsweise für ein Loginfenster oder ein Dialogfenster, da nur eine sehr eingeschränkte Auswahl an Komponenten zur Verfügung steht.

2.4 Mockup Screens

Mockup Screens ist ein kommerzielles Produkt von Igor Jese, dass zum Zeitpunkt der Erstellung des Berichtes in der Version 3.16 vorlag. Die Oberfläche besteht aus einem Fenster mit einem Menü und einer Toolbar. Unterhalb der Toolbar befindet sich die Arbeitsfläche in der linken und eine Baumstruktur des Projektes in der rechten Spalte.

Das Menü besteht aus den Punkten *File*, *Scenario*, *Screen* und *Help*. Die Einträge des Menüpunktes *File* sind bis auf die Funktionen *Save as*, *Rename Project*, *Export Project*, *Recent files* und *Exit*, bereits in der Toolbar vorhanden und sollten daher nicht erneut im Menü verfügbar sein um Redundanzen zu vermeiden. Sie dienen dem Neuanlegen, dem Öffnen und Speichern von Projekten, sowie der Einstellung von weiteren Optionen bezüglich des Exports, der Editierung und der Darstellung von Projekten innerhalb einer Slideshow. Mittels der Exportfunktion kann entweder das gesamte Projekt oder ein einzelnes Szenario in HTML exportiert werden. Die Exportfunktion ist jedoch nur anwendbar, wenn sich der Benutzer bereits registriert hat. Desweiteren ist es fraglich, ob die Funktion *Recent files* einen zusätzlichen Nutzen für den Benutzer beinhaltet, da bereits eine Funktion zum Öffnen der Dateien besteht. Die Menüpunkte *Scenario* und *Screen*

weisen einen sehr hohen Grad an Redundanz auf, da ihre Funktionen bis auf *Rename Scenario* beziehungsweise *Screen* und *Edit Screen*, bereits in der Toolbar realisiert sind. Der Menüpunkt *Help* beinhaltet außer einer Hilfedatei und den Informationen über den Entwickler der Software, zudem Einträge zur Registrierung, zum Onlinekauf des Produktes und den Besuch der MockupScreens Website. Da der Benutzer bereits bei jedem Start der Software zum Kauf und zur Registrierung aufgefordert wird, bieten sie dem Benutzer keinen zusätzlichen Nutzen. Die Hilfedatei selbst liegt als PDF vor und beinhaltet zu den wichtigsten Funktionen der Software Beispiele und Anleitungen zu ihrer Erstellung, welche sowohl natürlichsprachlich als auch graphisch erläutert werden. Die Menüeinträge von Mockup Screens weisen einerseits einen hohen Grad an Redundanz auf und andererseits einige Funktionen, welche dem Benutzer keinen zusätzlichen Nutzen bringen. Da die Toolbar durch den Benutzer nicht verändert werden kann, wäre eine redundanzfreie Realisierung des Menüs durchaus möglich, beziehungsweise ein vollständiger Verzicht auf die Menüstruktur denkbar.

Die Anlegung eines neuen Projektes kann entweder über den entsprechenden Eintrag im Menüpunkt *File*, oder über die Verwendung des *New Project-Ikons* erfolgen, sowie über die Tastenkombination *Ctrl-N*¹. Nachdem ein neues Projekt angelegt wurde, tritt jedoch keine Änderung innerhalb des *Clipboard*s beziehungsweise der Arbeitsfläche auf. Ebenso verhält es sich nach einem Speichervorgang mit der zugehörigen Baumstruktur. In der Baumstruktur wird das gespeicherte Projekt weiterhin als *New Project* bezeichnet, auch dann, wenn bei der Speicherung eine andere Bezeichnung vergeben wurde. Dies kann nur durch eine explizite Umbenennung innerhalb der Baumstruktur behoben werden.

Der Benutzer muss, nachdem er ein neues Projekt angelegt hat, dem Clipboard einen Screen hinzufügen, entweder über das entsprechende Icon in der Toolbar, oder über den Eintrag *New Screen* im Menüpunkt *Screen*. Da dies stets notwendig ist, wäre es denkbar diesen Vorgang zu automatisieren. Mit Hilfe der Clipboardbuttons, kann ein Screen oder ein Szenario des Clipboard's kopiert und eingefügt werden. Ein markierter Screen kann zudem von einem Szenario in ein anderes Szenario verschoben oder als Template verwendet werden, um ihn in Folgeprojekten weiterverwenden zu können. Die Templates bestehen aus einer Reihe von vorgefertigten Screens, welche einem Szenario hinzugefügt werden können. Die in diesen Screens enthaltenen Elemente können vom Benutzer frei editiert werden. Zu beachten ist, dass bevor ein Screen erstellt werden kann, ein Szenario vorhanden sein muss. Hat der Benutzer beispielsweise das Szenario in der Baumstruktur gelöscht, ist es, sofern kein weiteres Szenario außer dem *Main Szenario* existiert, nicht

¹Bei Verwendung einer deutschsprachigen Tastatur entsprechend *Strg-N*

möglich, einen Screen anzulegen. Um einen vorhandenen Screen im *Screen Editor* bearbeiten zu können, muss der Benutzer entweder den Eintrag *Edit Screen* im entsprechenden Menüpunkt verwenden oder einen Doppelklick auf den Screen ausführen. Im *Screen Editor* stehen dem Benutzer Standardelemente, wie Buttons, Listen und Tabellen zur Verfügung, drei einfache und drei gelabelte Textelemente, sowie vier Webpage Elemente. Eine Übersicht über die verfügbaren Elemente gibt die Abbildung 2.4.

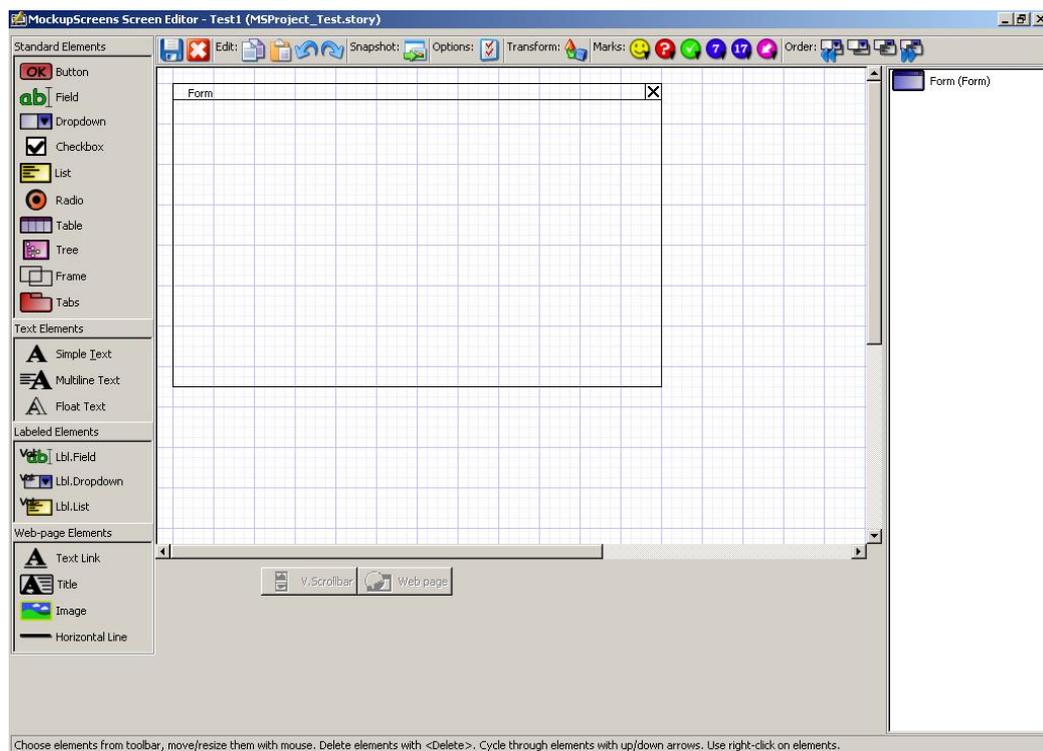


Abbildung 2.4: Verfügbare Elemente im Screen Editor

Die Elemente können innerhalb des Screens verschoben werden und sind untereinander frei kombinierbar. Eine Skalierung ist jedoch bei einigen Elementen nicht möglich, da zwar beispielsweise das eine Checkbox umgebende Feld vergrößert werden kann, sich die Größe der Checkbox selbst jedoch nicht ändert.

Am linken Fensterrand befindet sich zudem eine Auflistung der bisher hinzugefügten Elemente, welcher zu Beginn nur aus dem Screen selbst besteht. In der Übersicht können die vorhandenen Elemente sowohl ausgewählt, als auch, mit Ausnahme des Screen, gelöscht werden. Um die verwendeten Elemente zu bezeichnen, oder bei Textelementen Texte hinzuzufügen, muss der Benutzer das jeweils gewünschte in die Felder unterhalb der Ar-

beitsfläche eintragen. Die Eingabefelder variieren in ihrer Anzahl und Funktion bezüglich eines gewählten Elements. Beispielsweise kann der Benutzer bei dem Element Tab für Karteireiter, weitere Karteireiter hinzufügen, indem er diese im entsprechenden Eingabefeld, unter die beiden bereits vorhandenen, einträgt und anschließend die Arbeitsfläche markiert. Zudem befinden sich unterhalb der Arbeitsfläche ein Button um dem Screen, oder einigen seiner Elementen, wie zum Beispiel einer Tabelle, eine Scrollbar hinzuzufügen, sowie ein Button Webpage, der unterhalb der Menüleiste des Screens, diesem eine Browser spezifische Toolbar hinzufügt. Bei diesen beiden Elementen ist jedoch weder eine Skalierung noch eine Verschiebung möglich. Möchte der Benutzer bei der Gestaltung seines Screens jedoch auf die Verwendung eines Menüs verzichten, so hat er stets eine Leerzeile zwischen dem Fenstertitel und den weiteren Fensterelementen. Eine individuelle Gestaltung der klassischen Webpageelemente ist ebenfalls nicht möglich.

In der Toolbar oberhalb der Arbeitsfläche stehen dem Benutzer Marks zur Verfügung. Diese können beispielsweise bei der Erstellung von Dialogboxen, Fehler- oder Warnmeldungen gezielt eingesetzt werden, um die Aufmerksamkeit des Betrachters darauf zu lenken. Bei einer Warnmeldung könnte hierzu ein Ausrufezeichen verwendet werden. Die Marks dienen jedoch nicht nur zur Aufmerksamkeitssteuerung, sondern können auch zur Nummerierung der erstellten Screens innerhalb eines Szenarios eingesetzt werden, da ein solches meist aus einer Folge von Screens besteht, insbesondere bei der Gestaltung von Webauftritten. Zudem können die Marks auch kommentiert werden, so dass bei einer Präsentation eines erstellten Szenarios, diese unterstützend eingesetzt werden können, sowohl für den Präsentierenden als auch für sein Auditorium. Innerhalb dieser Toolbar stehen dem Benutzer, neben den Marks, der Speicherfunktion, den Undo- und Redobuttons, sowie der Clipboardbuttons, noch ein Transformationsbutton, ein Schnappschußbutton, der bereits erwähnte Optionsbutton und vier Buttons zur Festlegung der Reihenfolge bei sich überlagernden Elementen, zur Verfügung. Mit Hilfe des Transformationsbuttons, kann der Benutzer ein bereits beschriftetes Element in seinem Screen in ein anderes Element seiner Wahl überführen ohne dass die Beschriftung erneut eingegeben werden muss. Hierzu muss der Benutzer das zu transformierende Element markieren, den Transformationsbutton drücken und aus der Dropdown Liste das gewünschte Element auswählen. Zu beachten ist jedoch, dass es bei einigen Transformationen zu Informationsverlust kommen kann, insbesondere wenn man ein Textelement, welches aus mehreren Zeilen besteht, beispielsweise in eine Checkbox transformiert.

Mittels des Schnappschußbuttons können Bilder im Gif oder JPEG Format von einem Screen oder Szenario erstellt werden. Der Benutzer muss somit nicht auf einen Screens-

hot zurückgreifen, welchen er mit zusätzlicher Software nachbearbeiten müsste, sondern kann die Bilder direkt verwenden, beispielsweise in einer Präsentation.

MockupScreens bietet dem Benutzer eine gute Unterstützung bei der Erstellung von Webseiten, Interaktionselementen und Oberflächen, um nur einige Anwendungen zu nennen. Zudem bietet es dem Benutzer die Möglichkeit, seine erstellten Projekte in einer Slideshow zu präsentieren, so dass der Benutzer hier nicht auf weitere Softwareprodukte zurückgreifen muss.

2.5 Denim

Denim ist ein nichtkommerzielles Softwareprojekt der Universität Washington² und lag zu dem Zeitpunkt der Erstellung dieses Berichtes in der Version 2.0 vor.

Die Oberfläche von Denim besteht aus einem Menü, einer Werkzeugleiste, einer Arbeitsfläche, sowie einem Zoomslider. Das Menü ist, im Gegensatz zu den bisher vorgestellten Softwareprodukten, kreisförmig realisiert worden. Um das Menü öffnen zu können, muss der Benutzer den Menükreis in der linken oberen Ecke der Oberfläche aktivieren oder die rechte Maustaste, beziehungsweise die Taste am Grafikstift, gedrückt halten. Das Menü erscheint zentriert in der Arbeitsfläche und besteht aus sechs Einträgen. Die Einträge *File*, *Edit* und *Help* sind im oberen Halbkreis realisiert und die Einträge *View*, *Insert* und *Context* in der unteren Hälfte des Kreises. Bei Anwahl eines Menüpunktes öffnet sich ein neuer Menükreis mit den jeweiligen Menüeinträgen, der den ursprünglichen Menükreis teilweise verdeckt. Existieren weitere Einträge zu einem Eintrag, so ist dies mit einer Pfeilspitze gekennzeichnet. Die Menüstruktur besitzt mindestens zwei Ebenen und weist maximal drei Hierarchieebenen, beispielsweise im Menüpunkt *File*, auf. In diesem stehen dem Benutzer Standardfunktionen wie Speichern, Drucken oder Öffnen von Projekten zur Verfügung, sowie eine Funktion *Component*, mit welcher der Benutzer eigene Komponenten erstellen, bearbeiten, löschen, importieren oder exportieren kann. Im Menüpunkt *View* stehen dem Benutzer zudem vier Zoomfunktionen zur Verfügung, wovon zwei der Funktionen, *Zoom In* und *Zoom Out*, lediglich Änderungen an der Einstellung des Zoomsliders bewirken. Wurde eine der beiden Funktionen verwendet, schließt sich das Menü wieder und der Benutzer müsste das Menü erneut öffnen, sofern er entweder durch *Zoom In* den Detailgrad erhöhen, oder mittels *Zoom Out*, diesen verringern möchte.

Da der Zoomslider durch den Benutzer nicht in seiner Funktion verändert oder geschlos-

²ehemals Berkeley

sen werden kann, ist die zusätzliche Realisierung dieser beiden Funktionen innerhalb des Menüs redundant und in Bezug auf ein effektives Ein- und Auszoomen ungeeignet, da sich das Menü nach einmaliger Verwendung von einer der Funktionen sofort schließt. Die anderen beiden Funktionen dienen zum einen der Zentrierung der Arbeitsfläche und zum anderen kann sich der Benutzer, zur Wahrung der Orientierung, wenn er zum Beispiel mehrere Seiten erstellt hat, eine stark verkleinerte Ansicht der Arbeitsfläche einblenden lassen, in welcher sich der Cursor als rotes Kreuz darstellt. In dem Menüpunkt *Edit* stehen dem Benutzer klassische Funktionen wie Einfügen, Ausschneiden, Löschen und Kopieren, sowie eine Undo und Redo Funktion, zur Verfügung. Er beinhaltet zudem die Funktion *Options*, zur Einstellung der Seitengröße, des Zeitintervalls der automatischen Speicherung und der Feinheit des Gitternetzes. Der Menüpunkt *Insert* stellt dem Benutzer drei Funktionen zur Verfügung, mittels derer er eine neue Seite, einen Text oder Komponenten einfügen kann. Der Menüpunkt *Context* hingegen, wird dynamisch angepasst. Dies bedeutet, wenn der Benutzer beispielsweise eine Checkbox markiert hat und die rechte Maustaste gedrückt hält, öffnet sich das Menü und anstelle des Eintrags *Context* erscheint nun der Eintrag *Checkbox Instance*, mit dem er zum Beispiel festlegen kann, ob die Checkbox angewählt ist oder nicht. In dem Menüpunkt *Help* befindet sich neben der Hilfefunktion, welche als *Content* bezeichnet ist, zudem Informationen über die Entwickler des Projektes, sowie die Version in der es vorliegt.

Die Hilfefunktion öffnet sich in einem separaten Fenster und besteht aus einer Toolbar, einer Spalte mit drei Karteireitern, welche das Inhaltsverzeichnis, den Index und eine Suchfunktion beinhalten, sowie dem Inhalt selbst. Die Toolbar bietet die Möglichkeit, einzelne Einträge für den Druck einzurichten und auszudrucken. Die Navigation innerhalb der Hilfe kann sowohl über die Vor- und Zurückbuttons in der Toolbar erfolgen, als auch über das Inhaltsverzeichnis, durch Anwählen der entsprechenden Einträge, oder über den Index, in welchem zusätzlich zur Suche selbst, noch eine Indexsuche realisiert wurde. Der Aufbau der Hilfe untergliedert sich grob in eine Einführung in die Basisfunktionen von Denim und einer Referenz mit zusätzlichen Hinweisen. Beide Kategorien enthalten eine kurze beispielgestützte sprachliche Erläuterung der Funktionen, welche zusätzlich noch graphisch visualisiert sind. Anzumerken ist, dass sich der Benutzer zunächst mit der in Denim verwendeten Terminologie in der Hilfe befassen muss, da diese in den weiterführenden Einträgen als bekannt vorausgesetzt wird.

Der Zoomslider beinhaltet fünf Zoomübersichten, die entweder über Verschieben des Sliders, oder über das Menü durch Ein- beziehungsweise Auszoomen, ausgewählt werden können. In der Einstellung *Overview*, werden die erstellten Seiten als Thumbs dargestellt,

was nur einen groben Überblick ermöglicht. In der *Site Map* Übersicht werden sowohl die Seiten, als auch ihre Namen angezeigt. Die Übersicht *Storyboard* hingegen eignet sich um Verknüpfungen zwischen den Seiten zu erstellen. Mit Hilfe der *Pageübersicht*, kann sich der Benutzer eine Seite in ihrer vollen Größe anzeigen lassen oder, über die *Detailübersicht*, einzelne Komponenten der erstellten Seite.

Die Anlegung einer Seite kann auf dreierlei Art erfolgen, über den Eintrag Page im Menüpunkt Insert, über das Schreiben des Seitentitels mit dem Stift oder über das Zeichnen eines Rechtecks. Der Benutzer muss für die letztgenannte Variante ein möglichst geradliniges und geschlossenes Rechteck zeichnen, damit dies als eine neue Seite angelegt wird. Hierzu benötigt man folglich einige Übung im Umgang mit dem Grafiktablett beziehungsweise mit der Maus, sofern ein solches nicht zur Verfügung steht. Um der neu erstellten Seite Inhalte hinzuzufügen, kann entweder das Menü oder aber die Toolbar genutzt werden. Anzumerken ist jedoch, dass einige Werkzeuge der Toolbar nicht in den Ansichten *Overview* und *Site Map* verwendet werden können. Dies sind die Werkzeuge für den Radiobutton, die Checkbox, den Pushbutton, das Textfeld, die Listbox und die Combobox. Eine Übersicht über den Oberflächenaufbau, die verfügbaren Werkzeuge, sowie ein Beispiel einer mit Denim erstellten Seite, veranschaulicht die Abbildung 2.5.

Die Toolbar selbst besteht, neben den bereits genannten Werkzeugen, noch aus dem Stift, mit welchem die Seiten angelegt und beschrieben werden können, aus einem Radierer, um einzelne Komponenten oder ganze Seiten zu löschen, einem Füller zur Gruppierung der Komponenten, sowie einem Handwerkzeug zur Positionierung der Seiten. Nach dem Start von Denim ist standardmäßig das Stiftwerkzeug aktiv. Um ein anderes Werkzeug zu wählen, muss der Benutzer nicht, wie intuitiv zu erwarten wäre, ein anderes Werkzeug auswählen, sondern zuerst das derzeit aktive in die Toolbar zurücklegen, bevor er ein anderes nehmen kann. Problematisch daran ist, dass die Werkzeuge zudem keine feste Reihenfolge in der Toolbar besitzen, sondern die Möglichkeit besteht, ein Werkzeug beliebig in ihr abzulegen. Die Werkzeuge können zwar durch den Benutzer frei innerhalb der Toolbar positioniert werden, dies wird jedoch von der Software nicht dauerhaft übernommen, so dass sich die Toolbar bei einem erneuten Start, wieder in ihrem ursprünglichen Zustand befindet. Die mit den Werkzeugen erstellbaren Komponenten, wie zum Beispiel die Listbox, können sowohl skaliert als auch verschoben werden. Um ein Objekt verschieben zu können muss dies, bei Verwendung der Maus, zunächst über die rechte Maustaste ausgewählt werden und anschließend mit gedrückter linker Maustaste gezogen werden. Ähnlich verhält es sich bei der Verwendung des Grafikstiftes. Der Benutzer muss, um das Objekt wählen zu können, die Taste am Stift über dem Objekt drücken und kann dieses

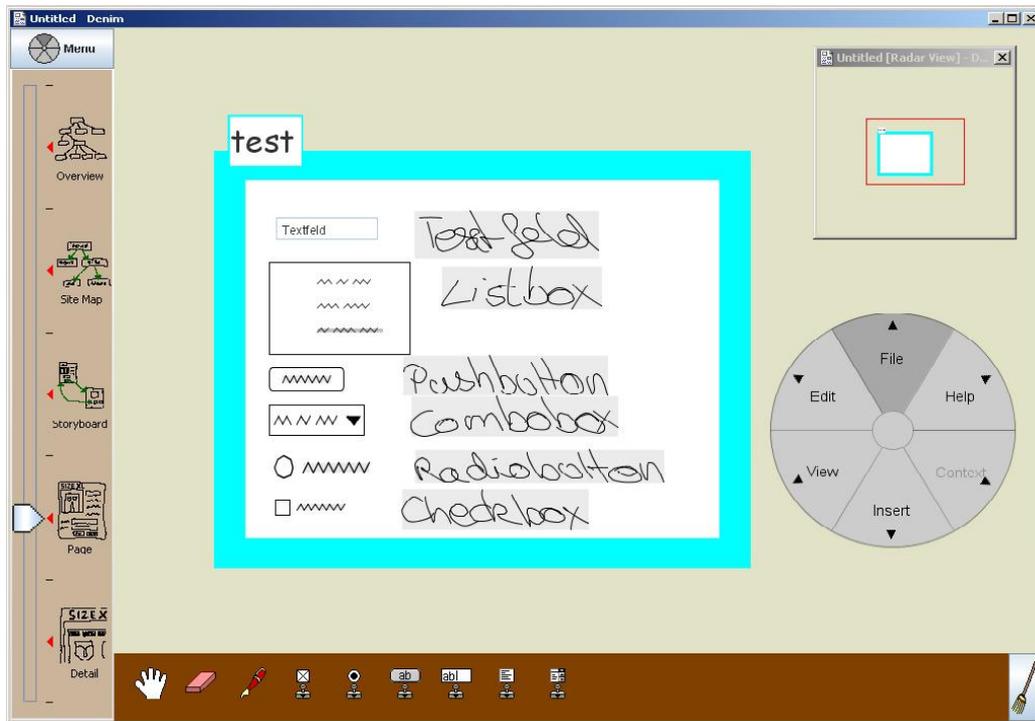


Abbildung 2.5: Übersicht über den Aufbau und die verfügbaren Werkzeuge

dann mittels der Stiftbewegung verschieben.

Denim ermöglicht einem Benutzer eine einfache erste Gestaltung von Webseiten und Oberflächen mit ihren grundlegendsten Elementen in der frühen Entwurfsphase und kann als ein guter Ersatz für Stift und Papier betrachtet werden. Es bietet dem Benutzer zudem die Möglichkeit, erstellte Seiten in einem Browser laufen zu lassen, um beispielsweise die Verknüpfungen der Seiten zu überprüfen.

2.6 Vergleich

Die in den vorangegangenen Abschnitten vorgestellten Softwareprodukte werden im weiteren Verlauf kurz einander gegenübergestellt und abschließend bewertet.

Die Softwareprodukte SmartDraw und MS Visio weisen in ihrem Oberflächenaufbau starke Ähnlichkeiten auf, insbesondere bei der Positionierung von Hilfe, Arbeitsfläche und Komponentenauswahl. Zudem befinden sich beim Start der beiden Produkte in der linken Spalte die Gestaltungskategorien und in der mittleren Spalte die entsprechenden Beispielvorlagen. Nach Auswahl einer Kategorie befinden sich die, durch eine betitelte Schalt-

fläche repräsentierten, vordefinierten Komponentengruppen, jeweils in der linken Spalte der Oberfläche. Bei beiden ist jeweils nur die aktive Auswahl sichtbar und die Auswahlen der verschiedenen Kategorien sind untereinander kombinierbar. Ebenso wurde das Hinzufügen von Komponenten auf die Arbeitsfläche bei beiden Produkten über das Drag und Drop Prinzip realisiert. Unterschiede in Bezug auf die Oberflächengestaltung finden sich im oberen Bereich. So besteht dieser Bereich bei MS Visio aus einem sehr umfangreichen Menü und einer Toolbar, wohingegen sich bei SmartDraw dieser Bereich in fünf Schaltflächengruppen aufteilt, die teilweise ein eigenes Menü unterhalb ihrer Buttons besitzen. Das Menü selbst ist bei MS Visio in Verbindung mit der Toolbar sehr redundant, da alle in der Toolbar verfügbaren Funktionen auch innerhalb des Menüs realisiert wurden, welches dementsprechend sehr unübersichtlich ist. Ähnlich verhält es sich bei MockupScreens. Dieses Produkt besitzt zwar einen deutlich geringeren Funktionsumfang als MS Visio oder SmartDraw, weist jedoch ebenso wie MS Visio einen hohen Grad an Redundanz bezüglich seines Menüs in Verbindung mit seiner Toolbar auf. Bei SmartDraw hingegen befinden sich innerhalb der Menüpunkte nur solche Einträge, welche nicht in dem jeweiligen Toolbar Bereich realisiert wurden. Dadurch wird die Übersichtlichkeit innerhalb der einzelnen Menüs bei SmartDraw gewahrt. Da nicht zu jedem Toolbarbereich ein Menü existiert, insbesondere in der Gruppe Design, kommt es hier zu einer inkonsistenten Menüverwendung. Meines Erachtens wäre hier ein genereller Verzicht auf die Verwendung von Menüs, beispielsweise durch den gezielten Einsatz von Multifunktions-tools, sinnvoller gewesen. Dennoch ist SmartDraw auch für einen unerfahrenen Benutzer sehr einfach zu bedienen, insbesondere aufgrund der zu jeder Komponente und zu jedem Werkzeug existierenden Popups mit kurzen Erläuterungen. Bei MS Visio hingegen benötigt selbst ein Benutzer der bereits Erfahrung im Umgang mit den MS Office Produkten besitzt eine längere Einarbeitungszeit, insbesondere aufgrund der Unübersichtlichkeit der Toolbar und des Menüs. Zudem existieren bei MS Visio zwar Popup Fenster mit der jeweiligen Bezeichnung der Komponente oder des Werkzeugs, diese enthalten jedoch nur bei den Komponenten eine sehr kurze Erläuterung. Ähnlich verhält es sich mit den Popup Fenstern bei Denim, GuiGenie und MockupScreens, da bei diesen zwar stets die Bezeichnung des jeweiligen Werkzeugs, beziehungsweise der jeweiligen Komponente zur Verfügung stehen, aber keine Erläuterungen bezüglich der Funktion der Werkzeuge. Ein weiterer Kritikpunkt bei MS Visio stellt die Hilfe selbst dar. Bei den anderen im Bericht vorgestellten Softwareprodukten befindet sich in jeder Hilfedatei sowohl eine sprachliche als auch eine graphische Erläuterung der grundlegenden Funktionen anhand von Beispielen, worauf bei der Hilfedatei von MS Visio verzichtet wurde.

Die Produkte GuiGenie und Denim wurden ebenso mit Menüstrukturen und Toolbars realisiert, wobei Denim, im Gegensatz zu den anderen Softwareprodukten, eine kreisförmige Menüstruktur besitzt. Bei GuiGenie und MockupScreens befindet sich die Toolbar, wie bei MS Visio, unterhalb des Menüs. Allerdings ist die Nützlichkeit der Toolbar bei GuiGenie fraglich, da sie nur aus drei Werkzeugen besteht, welche bereits innerhalb des Menüs vorhanden sind. Bei Denim hingegen, wurde sie am unteren Rand der Oberfläche realisiert. Das Hinzufügen von Komponenten geschieht sowohl bei Denim, als auch bei MockupScreens über anklicken der gewünschten Komponente und anschließendes anklicken der Arbeitsfläche. Anzumerken ist bei MockupScreens, dass der Benutzer, um Komponenten hinzufügen zu können, zunächst den Screen Editor öffnen muss. Bei GuiGenie hingegen ist das Hinzufügen von Komponenten sehr umständlich, da der Benutzer hierzu zunächst den entsprechenden Button drücken muss, um aus dem sich öffnenden Fenster eine Komponente auswählen zu können und zusätzlich seine Auswahl noch bestätigen muss. Allerdings bietet GuiGenie als einziges der vorgestellten Produkte, die Möglichkeit, sich direkt den Quellcode der damit erstellten Oberfläche generieren zu lassen, auch wenn dieser nicht durch den Benutzer direkt in GuiGenie bearbeitbar ist.

Die Produkte MockupScreens und GuiGenie bieten, im Gegensatz zu den anderen Softwareprodukten, jedoch keinerlei Unterstützung des Freihandzeichnens an, welches besonders für die frühe Entwurfsphase einer Oberfläche von Vorteil ist. Bei MS Visio und SmartDraw ist diese Funktion zwar vorhanden, eignet sich aufgrund ihrer Umsetzung jedoch nicht für die Erstellung erster Oberflächenentwürfe, im Gegensatz zu der Umsetzung bei Denim.

Die Produkte SmartDraw und MS Visio bieten dem Benutzer den umfangreichsten Funktionsumfang und eine große Auswahl an vordefinierten Elementen. Die anderen drei Produkte stellen dennoch eine Alternative innerhalb der verschiedenen Entwurfs- und Entwicklungsphasen der Oberflächengestaltung zu ihnen dar. Zudem sollte auch der Aspekt der Anschaffungskosten der Produkte nicht unberücksichtigt bleiben. So sind die Softwareprodukte Denim und GuiGenie kostenlos verfügbar, wohingegen eine Einzelplatzlizenz von SmartDraw 197\$, bei MS Visio Professional bereits 495€ und bei MockupScreens 79\$ kostet. Die kommerziellen Softwareprodukte bieten dem Benutzer die Möglichkeit sich eine kostenlose, zeitlich begrenzte, Trialversion herunterzuladen um diese zuvor testen zu können.

Abschließend sei darauf hingewiesen, dass jedes Produkt für sich nicht alle Entwicklungsphasen einer Oberflächengestaltung abdeckt und der Benutzer letztlich auf eine Kombination von Softwareprodukten oder zusätzlich auf Stift und Papier, zurückgreifen muss.

Kapitel 3

Entwicklung der Benutzungsoberfläche

In diesem Kapitel wird zunächst die Gestaltung und der Aufbau des Programms Scribble It vorgestellt und seine Umsetzung, aufbauend auf den gewonnenen Erkenntnissen des *State of the Art* Berichtes und in Anlehnung an die Thesen des WOB-Modells, begründet. Im weiteren Verlauf werden die einzelnen Klassen des Programms kurz beschrieben, um einen groben Überblick über diese zu geben.

3.1 Gestaltung und Aufbau der Benutzungsoberfläche

Der Gestaltung der Benutzungsoberfläche des Programms Scribble It lagen mehrere Kriterien zugrunde. Das Programm sollte intuitiv von einem Benutzer bedienbar sein, ohne dass eine längere Einarbeitungszeit, wie beispielsweise bei MS Visio oder MockupScreens, erforderlich ist. Hierbei war zu beachten, dass das Programm jedoch auch von einem erfahrenem Benutzer gleichsam effektiv nutzbar sein sollte. Ebenso sollte es dem Benutzer möglich sein, auf Altwissen, welches er sich im Umgang mit anderen Softwareprodukten, wie zum Beispiel mit Microsoft Paint erworben hat, zurückgreifen zu können. Zudem musste darauf geachtet werden, dass die Oberfläche des Programms einen klaren strukturellen Aufbau besitzt, welcher in seiner Anordnung unveränderlich bleibt um den Erwartungen eines Benutzers entsprechen zu können. Aus diesem Grund wurde die Oberfläche so konzipiert, dass unabhängig von der Verwendung eines Werkzeugs, diese in ihrer

Struktur bestehen bleibt, im Gegensatz zu dem Softwareprodukt MockupScreens, wo die Erstellung einer Oberfläche, in einem neuen Editor erfolgt, der in seinem Aufbau nur wenig mit der eigentlichen Oberfläche, nach dem Programmstart, gemein hat.

Ein entscheidendes Kriterium für den Oberflächenaufbau von Scribble It, war der vollständige Verzicht auf eine Menüstruktur. Die Verwendung von Menüs birgt zwar den Vorteil, viele Funktionen auf geringem Raum realisieren zu können, allerdings auch den entscheidenden Nachteil, dass es mit zunehmender Anzahl der realisierten Funktionen unübersichtlich wird. Ein weiterer Kritikpunkt bei Menüstrukturen ist, dass ein Benutzer teilweise bis zu drei Hierarchieebenen durchsehen muss, wie beispielsweise bei MS Visio, um die von ihm gewünschte Funktion benutzen zu können.

Eine Kombination aus einem Menü mit einer Toolbar, ist zwar durchaus eine Alternative zu einer reinen Menüstruktur, doch wie im *State of the Art* Bericht bereits festgestellt, führte dies bei den Produkten MockupScreens und MS Visio zu einem hohen Grad an Redundanz. Zwar wäre bei der Verwendung dieser Kombination eine redundanzfreie Realisierung möglich gewesen, doch stellt sich dann die Frage, welche Funktionen der Benutzer innerhalb des Menüs erwartet und welche in der Toolbar. Ein weiterer Nachteil an der Verwendung eines Menüs ist meines Erachtens, die fehlende Selbsterklärungsfähigkeit der Menüeinträge, da es bei der Verwendung *verbaler Handlungsanweisungen*¹ bei einigen Einträgen nicht klar ersichtlich ist, was sich hinter diesen verbirgt.

Als Beispiel sei auf den Eintrag *Vorgang* im Menüpunkt Shape bei MS Visio verwiesen. Dieser Eintrag besitzt eine weitere Hierarchieebene mit Funktionen, welche an die Mengenlehre angelehnt sind, um beispielsweise Schnittmengen von Shapeelementen zu erstellen. Dies geht jedoch aus der Bezeichnung *Vorgang* nicht hervor. Aus dem Verzicht auf eine Menüstruktur ergab sich jedoch das Problem, wie die Standardfunktionen, beispielsweise das Speichern von Projekten, visualisiert werden sollten. Eine Möglichkeit wäre gewesen, die Standardfunktion als Buttons in einer Toolbar zu realisieren und diese nebeneinander anzuordnen. Diese Möglichkeit wurde jedoch verworfen da dies bei zunehmender Anzahl der Funktionen, die Übersichtlichkeit dieses Bereiches stark verringert hätte. Zudem hätten sich aus dieser Variante weitere Probleme ergeben, wie beispielsweise der benötigte Raum, den diese Toolbar auf der Oberfläche beansprucht hätte.

Es musste daher eine Umsetzung gefunden werden, die wenig Raum auf der Oberfläche benötigte, eine Gruppierung der Standardfunktionen entsprechend den Erwartungen eines Benutzers erlaubte um die Übersichtlichkeit zu wahren, sowie kurze Bedienungswege für ein schnelles und effizientes Arbeiten ermöglichte. Aus diesen Überlegungen

¹Krause (2005), Seite 4f

heraus wurde ein Karteikartensystem verwendet. Jede Karteikarte beinhaltet eine Toolbar mit Schaltflächen für die klassischen Standardfunktionen, wovon jede einzelne über eine Kurzinformation verfügt, in der ihre Funktion benannt wird. Durch die Verwendung von Kurzinformationen soll die Selbsterklärungsfähigkeit des Programms und seiner Elemente erhöht werden. Das Karteikartensystem besteht derzeit aus den zwei Karteikarten Datei und Bearbeiten, welche die grundlegendsten Standardfunktionen beinhalten. Durch die Verwendung von Karteikarten, ist es nicht nur möglich dem Karteikartensystem weitere Karten hinzuzufügen, sondern auch die bestehenden Karten können um zusätzliche Funktionen erweitert werden.

In der Karteikarte Datei befinden sich die Funktionen zum Neu anlegen des Projektes, zum Speichern des Projektes, zum Speichern des Projektes unter einem vom Benutzer zu vergebenden Namen, zum Öffnen bestehender Projekte, zum Löschen von markierten Elementen auf der Arbeitsfläche und zum Drucken des Projektes. Die Funktionen *Neues Projekt anlegen* und *Speichern unter*, einem vom Benutzer zu vergebenden Namen, sind bereits eingeschränkt nutzbar. Die restlichen zur Verfügung gestellten Schaltflächen wurden für zukünftige Implementierungen vorgesehen. Die Einschränkung der bereits implementierten Funktionen bezieht sich darauf, dass bisher das Speichern des Projektes im JPG Format vorgesehen ist und eine Erweiterung auf weitere Dateiformate möglich ist. Zur Veranschaulichung der Karteikarte Datei dient die Abbildung 3.1.



Abbildung 3.1: Übersicht über die Funktionen der Karteikarte Datei

Die Karteikarte Bearbeiten beinhaltet weitere Standardfunktionen die einerseits das Arbeiten mit der Zwischenablage ermöglichen sollen und andererseits der Gestaltung des Hintergrundes und dem Einfügen von Grafiken auf der Arbeitsfläche dienen.

Zur Gestaltung des Hintergrundes stehen dem Benutzer zwei mögliche Gestaltungsvarianten zur Verfügung, welche über einen Dialog, in Anlehnung an den *visuellen Formalismus*², ausgewählt werden können. In diesem Dialog kann der Benutzer zwischen dem Einfügen einer Hintergrundgrafik oder dem Setzen einer Hintergrundfarbe auswählen. Wurde eine dieser Möglichkeiten gewählt und mit dem Button OK bestätigt, öffnet sich

²Krause (2005), Seite 9f

entsprechend der Auswahl entweder ein Farbauswahldialog oder ein Dateiauswahldialog. Der Benutzer kann sowohl in dem Dialog zur Auswahl eines Hintergrundes als auch in dem jeweiligen anderen Dialog, jederzeit diese Abbrechen, um keinen Hintergrund einzufügen. Die Abbildung 3.2 zeigt den Dialog zur Auswahl eines Hintergrundes.



Abbildung 3.2: Dialog zur Auswahl eines Hintergrundes

Möchte der Benutzer für seinen Entwurf eine Grafik verwenden, so kann er dies über die Schaltfläche *Grafik einfügen* realisieren. Die Grafik wird derzeit in der linken oberen Ecke der Arbeitsfläche, in einer vordefinierten Größe, eingefügt. Hier bietet sich ebenso die Möglichkeit das Programm um zusätzliche Funktionen, beispielsweise zur Verschiebung oder Skalierung einer Grafik, zu erweitern. Die Funktionen *Grafik einfügen* und *Hintergrundgrafik einfügen* unterstützen derzeit das Laden von Grafiken im JPG, GIF und PNG Format. Die Schaltflächen für die Standardfunktionen in der Karteikarte Bearbeiten, welche die Zwischenablage betreffen, wurden für zukünftige Implementierungen der Oberfläche hinzugefügt. Dies wären eine Rückgängig- und eine Wiederherstellen-Funktion für einen Arbeitsschritt sowie das Ausschneiden, Kopieren und Einfügen von auf der Arbeitsfläche befindlichen Elementen. Zur Veranschaulichung der Karteikarte Bearbeiten dient die Abbildung 3.3.



Abbildung 3.3: Übersicht über die Funktionen der Karteikarte Bearbeiten

Ein ebenso entscheidendes Kriterium für die Gestaltung der Oberfläche, war neben dem Verzicht auf die Verwendung von Menüstrukturen, die Unterbringung der Werkzeuge. Hier ergaben sich sowohl Anforderungen an die Gruppierung der Werkzeuge als auch

an deren Positionierung. Die Werkzeuge sollten in sinnvolle und den Erwartungen eines Benutzers entsprechende Kategorien aufgeteilt werden. In Anlehnung an das Projekt *De-nim* wäre in diesem frühen Entwicklungsstadium des Programms eine Realisierung aller Werkzeuge innerhalb eines Fensterbereichs möglich gewesen. Diese Möglichkeit wurde jedoch verworfen, da dieser Bereich bei zunehmender Anzahl der zur Verfügung gestellten Werkzeuge einerseits an Übersichtlichkeit verliert und andererseits zu viel Raum der Oberfläche benötigen würde.

Als Alternative wäre die Verwendung eines Fensters denkbar gewesen, welches die Werkzeuge beinhaltet. Einem Benutzer würde damit die Möglichkeit gegeben, dieses frei auf der Oberfläche zu positionieren oder es zu minimieren, um mehr Raum für die Arbeitsfläche zur Verfügung zu stellen. Diese Umsetzung findet sich beispielsweise in den Produkten Photoshop von Adobe und Gimp. Diese Variante hätte jedoch meines Erachtens weder die Anforderung der Ortskonstanz bezüglich der Fensterposition erfüllt, es sei denn, dass diesem Fenster eine feste Position auf der Oberfläche zugewiesen würde, noch hätte dies der Anforderung bezüglich einer sinnvollen Gruppierung und der Wahrung der Übersichtlichkeit genügt.

Aus diesem Grund wurde ein Palettensystem in Anlehnung an die Produkte SmartDraw und MS Visio entwickelt. Dieses Palettensystem ermöglicht eine Aufteilung der Werkzeuge in einzelne Kategorien, wobei stets nur der Inhalt einer Kategorie sichtbar ist. Dadurch ist es möglich, eine große Anzahl an Werkzeugen auf geringem Raum strukturiert unterzubringen und den zur Verfügung stehenden Raum effizient zu nutzen³. Zudem ermöglicht diese Umsetzung nicht nur das Hinzufügen weiterer Kategorien sondern auch das Hinzufügen von zusätzlichen Werkzeugen in bereits bestehende Kategorien, ohne die vorhandene Struktur zu verändern. Das Palettensystem besitzt derzeit zwei Kategorien, welche als Zeichnen und Textfeld bezeichnet sind.

Die Kategorie Zeichnen beinhaltet derzeit drei Werkzeuge. Diese wurden in Anlehnung an das WOB-Modell als *Funktionsobjekte* festgelegt, die auf die Zeichenfläche einwirken⁴. Dies sind ein Stiftwerkzeug zum Zeichnen von Freihandlinien, einen Radiergummi mit welchem bereits gezeichnete Komponenten gelöscht werden können sowie ein Mauswerkzeug um die beiden zuerst genannten Funktionen deaktivieren zu können.

Der Radiergummi wurde für spätere Implementierungen vorgesehen und besitzt zu diesem Zeitpunkt noch nicht die für ihn vorgesehene Funktionalität. Zur Veranschaulichung des Palettensystems und des Inhaltes der Kategorie Zeichnen dient die Abbildung 3.4.

³Krause (2005), Seite 10ff

⁴Krause (2005), Seite 9f



Abbildung 3.4: Inhalt der Kategorie Zeichnen

Die Kategorie Textfeld beinhaltet derzeit eine Schaltfläche für die spätere Realisierung eines Textwerkzeugs. Diese Kategorie sowie das beinhaltete Textwerkzeug wurden der Oberfläche bereits hinzugefügt, um einerseits die Funktionsweise des Palettensystem zu veranschaulichen und andererseits, um die zusätzlichen Einstellungsmöglichkeiten für dieses Werkzeug der Oberfläche bereits hinzufügen zu können. Die Abbildung 3.5 zeigt den derzeitigen Inhalt der Kategorie Textfeld.



Abbildung 3.5: Inhalt der Kategorie Textfeld

Im unteren Bereich der Oberfläche befinden sich die bereits erwähnten zusätzlichen Einstellungsmöglichkeiten für das Stiftwerkzeug und das Textwerkzeug. Wählt der Benutzer das Stiftwerkzeug an, so erscheint eine Schaltfläche mit zwei Buntstiften welche einen Farbauswahldialog zur Auswahl einer Linienfarbe öffnet und eine Combobox in welcher eine Linienstärke ausgewählt werden kann. Die Combobox zur Wahl einer Linienstärke wurde diesem Bereich für zukünftige Implementierungen hinzugefügt, so dass der Oberfläche keine weitere Komponente für das Stiftwerkzeug hinzugefügt werden muss. Die Abbildung 3.6 zeigt diesen Bereich bei Aktivierung des Stiftwerkzeugs.



Abbildung 3.6: Zusätzliche Einstellungen für das Stiftwerkzeug

Ähnlich verhält es sich mit den Elementen für das Textwerkzeug. Bei Wahl

des Textwerkzeugs werden jeweils eine Combobox zur Einstellung einer Schriftart und einer Schriftgröße eingeblendet sowie eine Schaltfläche die einen Farbauswahldialog zur Auswahl einer Textfarbe öffnet. Die Einstellungsoptionen für das Textwerkzeug wurden einerseits für spätere Implementierungen diesem Bereich hinzugefügt und andererseits zur Verdeutlichung des Verhaltens von diesem Bereich. Die Abbildung 3.7 zeigt die Einstellungsoptionen für das Textwerkzeug bei seiner Aktivierung.



Abbildung 3.7: Zusätzliche Einstellungen für das Textwerkzeug

Stünden beispielsweise die Einstellungsoptionen aller Werkzeuge stets dem Benutzer zu jedem Zeitpunkt zur Verfügung, würden sich dadurch gleich mehrere Probleme ergeben. Einerseits wäre eine *effiziente Platzverwendung*⁵ dieses Bereichs nicht mehr möglich und andererseits bliebe die Anforderung an die Gruppierung zusammengehöriger Einstellungen zu ihren Werkzeugen unerfüllt. Zudem ergibt sich das Problem, dass der Benutzer versucht sein könnte, Einstellungen zu verwenden die für das derzeit aktive Werkzeug nicht geeignet sind. Hierdurch kann es unter Umständen zu Fehlern innerhalb des Programms kommen, welche schlimmstenfalls zu einem Neustart des Programms führen. Um dieser Problematik bereits im Vorfeld begegnen zu können, wurde dieser Bereich derart konzipiert, dass die zusätzlichen Einstellungsmöglichkeiten nur bei Verwendung des entsprechenden Werkzeugs für den Benutzer sichtbar werden. Dadurch soll sichergestellt werden, dass dem Benutzer stets nur diejenigen Einstellungen zur Verfügung stehen, welche auch auf das jeweilige Werkzeug anwendbar sind. Ein weiterer Vorteil dieser Realisierung besteht nicht nur in der effizienten Raumausnutzung, sondern auch in ihrer Erweiterbarkeit durch zukünftige Implementierungen, da in diesem Bereich weitere Einstellungselemente für zukünftige Werkzeuge hinzugefügt werden können.

Für die spätere Implementierung einer Zoomfunktion wurde am rechten Rand der Oberfläche, in Anlehnung an das Denim Projekt, bereits die Komponenten für die Realisierung einer Zoomfunktion hinzugefügt. Hierfür wurden drei verschiedene Ansichten zur Verfügung gestellt. Dies sind eine Gesamtübersicht, eine Seitenansicht und eine Detailansicht. Zur Auswahl einer der Ansichten soll später der hinzugefügte Schieberegler dienen. Die Abbildung 3.8 zeigt den Zoombereich des Programms.

Die Arbeitsfläche des Programms befindet sich im Zentrum der Oberfläche. Wurde ein

⁵Krause (2005), Seite 10ff



Abbildung 3.8: Darstellung des Zoombereichs

neues Projekt angelegt, erscheint in diesem Bereich eine weiße Zeichenfläche. Auf dieser Fläche kann der Benutzer derzeit einen ersten Oberflächenentwurf, unter Verwendung des Stiftwerkzeugs und dem Einfügen eines Hintergrunds seiner Wahl, gestalten. Die Abbildung 3.9 zeigt die Oberfläche des Programms in seiner Gesamtheit nach dem Anlegen eines neuen Projektes und der Aktivierung des Stiftwerkzeugs.

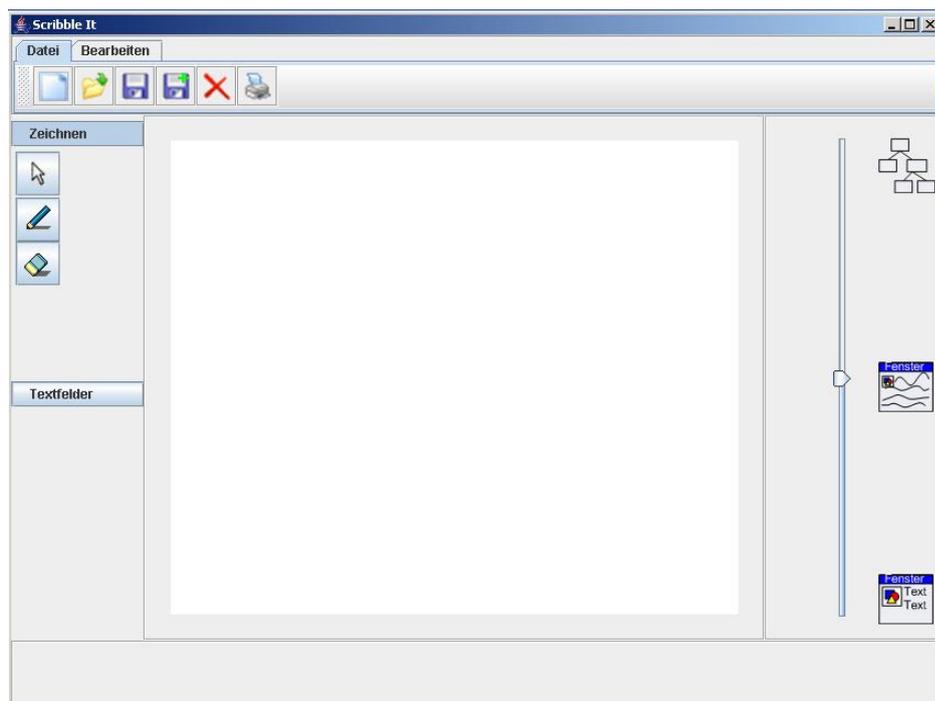


Abbildung 3.9: Oberfläche von Scribble It

Die weiße Fläche entspricht im Sinne des WOB-Modells⁶ einem Objekt, auf welches

⁶Krause (2005), Seite 8f

mittels der Funktionsobjekte, beispielsweise der Werkzeuge oder der Schaltfläche zum Einfügen einer Grafik, eingewirkt werden kann.

3.2 Programmstruktur

In den folgenden Unterabschnitten werden die Klassen des Programms kurz vorgestellt, um einen groben Überblick über diese zu vermitteln. Für zukünftige Implementierungen und Erweiterungen steht neben der Kommentierung des Quellcodes selbst, noch eine zusätzliche Dokumentation zur Verfügung, in welcher die Methoden der Klassen näher beschrieben werden.

3.2.1 Die Klassen ScribbleUI und Main

Die Klasse ScribbleUI wurde von der Klasse JFrame abgeleitet und umfasst alle für die Oberfläche bisher erforderlichen Komponenten. In dem Konstruktor der Klasse, in welchem beispielsweise die Größe des Fensters, sowie sein Titel festgelegt werden, werden zudem noch drei Methoden aufgerufen. Dies sind die Methoden `initAllAreas()`, `initLayout()` und `initListener()`. Innerhalb der Methode `initLayout()` werden die jeweiligen Layouts für die fünf Bereiche der Oberfläche, sowie ihrer Komponenten initialisiert. Innerhalb der Methode `initListener()` werden die jeweils benötigten `ActionListener` für die einzelnen Schaltflächen initialisiert. Die Methode `initAllAreas()` hingegen, ruft sechs Methoden auf. Diese Methoden initialisieren die jeweiligen Komponenten der fünf Bereiche der Oberfläche sowie des Dialogs zur Auswahl eines Hintergrunds. Diese Aufteilung bietet den Vorteil, dass bei zukünftigen Erweiterungen oder Änderungen der Oberfläche, die benötigten Initialisierungen, beziehungsweise Änderungen, in die jeweils dem Bereich zugeordnete Methode eingefügt werden können. Zudem wurde so die Lesbarkeit des Quellcodes deutlich erhöht. Die Klasse ScribbleUI enthält zudem mehrere `ActionPerformed` Methoden derjenigen Schaltflächen, von denen sich meines Erachtens, eine Implementierung ihrer Funktion in eigenen Klassen nicht angeboten hat.

Die Klasse Main besitzt nur eine einzige Methode, `main()`, in welcher eine Instanz der Klasse ScribbleUI erzeugt wird und das Programmfenster auf sichtbar gesetzt wird.

3.2.2 Die Klassen Scribblesheet und LineColor

Die Klasse Scribblesheet wurde von der Klasse JPanel abgeleitet und enthält zwei innere Adapterklassen vom Typ MouseListener beziehungsweise MouseMotionListener. Die inneren Klassen dienen dazu, die jeweiligen Positionen der Maus beim Freihandzeichnen zu ermitteln und in einem Polygon zu sichern. Dieses Polygon wird einem Vektor hinzugefügt, aus welchem bei einem neu zeichnen der Oberfläche die Mauspositionen wieder ausgelesen werden können und die Linie erneut gezeichnet werden kann. Würden diese Positionen nicht gesichert, könnte der Benutzer stets nur einen einzigen Linienzug zeichnen, da bei jedem neuen ansetzen des Stiftes ein Ereignis ausgelöst wird, welches ein neu zeichnen der Oberfläche verursacht. Die Verwendung des Vektors bietet zudem die Möglichkeit, alle gezeichneten Linien, von der Oberfläche löschen zu können, indem der Inhalt des Vektors gelöscht wird. Die Klasse Scribblesheet enthält zudem eine Methode mittels derer eine vom Benutzer gesetzte Linienfarbe übernommen werden kann, sowie zwei weitere Methoden, die zum aktivieren beziehungsweise deaktivieren des Freihandzeichnens dienen. In der Klasse LineColor wird der für die Linienfarbe vorgesehene Farbauswahl-dialog erzeugt und die vom Benutzer gewählte Farbe mittels einer Parameterübergabe an die entsprechende Methode der Klasse Scribblesheet übergeben.

3.2.3 Die Klassen ImagePaste, Imagecomponent und Backgroundcomponent

Die beiden Klassen Imagecomponent und Backgroundcomponent sind von der Klasse JPanel abgeleitet und beinhalten jeweils drei Methoden gleicher Funktionalität. Die erste Methode dient dazu, eine vom Benutzer gewählte Grafikdatei als BufferedImage einzulesen, dieses in einem Vektor zu sichern und auf die benötigte Größe zu skalieren. In der zweiten Methode wird das skalierte Bild auf die entsprechende Komponente gezeichnet. Die letzte Methode dient zum Löschen des Bildes, indem hier wieder der Inhalt des Vektors gelöscht wird. Die Klasse Imagecomponent dient dem Einfügen von Grafiken, welche sie jeweils von der Klasse ImagePaste durch eine Parameter übergeben bekommt. Dementsprechend wird in der Klasse ImagePaste der hierfür vorgesehene Dateiauswahl-dialog erzeugt und die vom Benutzer gewählte Grafikdatei dem zu übergebenden Parameter zugewiesen. Die Klasse Backgroundcomponent dient hingegen dem Einfügen von Hintergrundgrafiken, die sie von der Klasse ScribbleUI übergeben bekommt.

3.2.4 Die Klassen ImageFilter, SaveImageFilter und Utils

Die Klassen ImageFilter und SaveImageFilter wurden von der Klasse FileFilter abgeleitet und besitzen zwei Methoden. Die erste Methode prüft die Dateierweiterungen des gewählten Verzeichnisses und zeigt nur diejenigen Dateien an, deren Erweiterungen mit denen in der Klasse Utils festgelegten, übereinstimmen. Die zweite Methode liefert dem Benutzer nur die Information darüber, welche Dateierweiterungen akzeptiert werden. Die Klasse SaveImageFilter akzeptiert im Gegensatz zu der Klasse ImageFilter nur die Erweiterungen .jpg und .jpeg, da der Benutzer derzeit seine Projekte nur als JPG speichern kann. Diese Klassen wurden in Anlehnung an das Beispiel *FileChooserDemo* der Firma Sun erstellt. Dieses Beispiel ist unter der folgenden URL: <http://java.sun.com/docs/books/tutorial/uiswing/components/filechooser.html> zu finden.

3.2.5 Die Klasse Save

Die Klasse Save enthält einen Dateiauswahldialog mit welchem der Benutzer festlegen kann, wie das zu speichernde Projekt bezeichnet werden soll und wo es gespeichert werden soll. Hierzu werden die Grafikobjekte als BufferedImage der aus zwei JPanel bestehenden Zeichenfläche mit Hilfe eines Vektors in ein einziges BufferedImage geschrieben. Die Inhalte des BufferedImage werden in die neue Datei geschrieben und unter Verwendung der von Sun Microsystems bereitgestellten Codecs als JPG kodiert. Bei diesem Vorgang wird zusätzlich die Bildqualität und die Kompressionsrate festgelegt. Im Anschluß daran, wird die beschriebene Datei geschlossen und der temporär benötigte Speicher wieder freigegeben.

3.3 Ausblick

Das Programm Scribble It befindet sich derzeit in einem frühen Entwicklungsstadium und kann durch zukünftige Studien- und Diplomarbeiten beliebig erweitert werden. Für die meines Erachtens notwendigen Standardfunktionen, wurde die Oberfläche bereits so gestaltet, dass hier keine Änderungen vorgenommen werden müssen. Durch die weitgehende Trennung von Oberfläche und Anwendungslogik können diese Funktionen in eigenen Klassen realisiert werden. Zudem bietet die derzeitige Realisierung die Möglichkeit, um weitere Werkzeuge erweitert zu werden. Lediglich die Struktur des Programms ist derart vorgegeben, dass zusätzliche Werkzeuge entweder in die bereits bestehenden Kategorien des Palettensystems oder gegebenenfalls in eigene neue Kategorien eingefügt werden

sollten. Desweiteren wäre es wünschenswert, weitere oder zusätzliche Einstellungsmöglichkeiten zu den Werkzeugen in die entsprechenden Bereiche einzufügen, so dass sie nur für das zugehörige Werkzeug zur Verfügung stehen.

Literaturverzeichnis

[Krause 2005]

KRAUSE, Jürgen: *Visual Interaction on the Basis of the WOB-Model.*

<http://www.uni-koblenz.de/~krause/skripte/WOB06.pdf>,

Juni 2005

[Krüger 2006]

KRÜGER, Guido: *Handbuch der Java-Programmierung, 4.Auflage.*

Addison Wesley, München, 2006

[Microsystems 1995]

MICROSYSTEMS, Sun: *Creating a GUI with JFC/Swing.*

<http://java.sun.com/docs/books/tutorial/uiswing>,

1995 – 2007

[Ullenboom 2006]

ULLENBOOM, Christian: *Java ist auch eine Insel, 6.Auflage.*

Galileo Computing, 2006