

SOFTWAREPRAKTIKUM MEDIENINFORMATIK WS 05/06

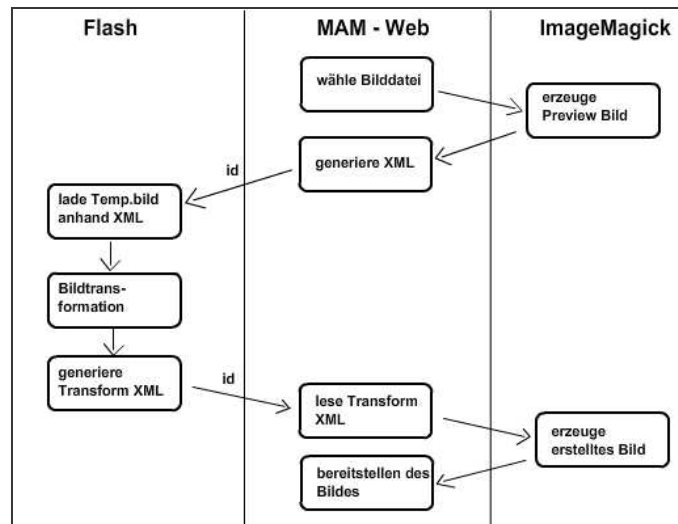
Dokumentation Online Bildeditor

Inhaltsverzeichnis

<u>DESIGN</u>	2
WORKFLOW ZUM BEARBEITEN EINES BILDES	2
META XML DATEI	2
TRANSFORM XML	3
KOMMUNIKATIONSSCHNITTSTELLEN	4
PERSISTENTE SPEICHERUNG DER TRANSFORMATIONEN	4
SHARED OBJECTS	4
FILEUPLOAD / DOWNLOAD	4
MULTILANGUAGE SUPPORT	5
<u>ACTIONSCRIPT</u>	6
CODEAUFBAU	6
Daten (data)	6
GUI (gui)	6
Manager (manager)	7
Styles (styles)	8
FILTER	8
<u>EINSATZ</u>	8
HTML	8
<u>IMPRESSUM</u>	9
<u>NACHWORT</u>	9

Design

Workflow zum Bearbeiten eines Bildes



Als Kommunikation zwischen dem Webserver und dem Client benötigen wir ein neutrales Datenformat, welches Metadaten des Bildes speichert und zwischen Client und Server über eine einfach GET/POST Abfrage ausgetauscht werden kann. Eine simple XML Struktur soll die Metadaten speichern.

Meta XML Datei

Die Meta XML Datei wird vom Webserver erstellt und als Parameter bei Aufruf der Bildbearbeitung übergeben. Flash kann sich dann aus dem XML den Pfad des temporär erzeugten Bildes nehmen und die übergebenen Metadaten als Initialzustand auf das Bild anwenden.

Meta XML:

```
<?xml verion="1.0" encoding = „iso-8859-1“>
<root>
  <tmppath> tempopath </tmppath>
  <imgid> imgid </imgid >
  <image_width> px </image_width>
  <image_height> px </image_height>
  <colorspace>rgb<colorspace>
</root>
```

Die Rückübergabe des Bearbeiteten Bildes erfolgt ebenfalls in XML. Die XML Datei beinhaltet die einzelnen Transformationsschritte. Die Übergabe erfolgt über den Befehl „sendAndLoad“ des XML Objekts.

```
XML.sendAndLoad(url:String, targetXMLObject:XML) : Void
```

Als Rückgabewert kann das aufgerufene Script einen Boolean senden, der die Erfolgreiche Bearbeitung des Bildes bestätigt.

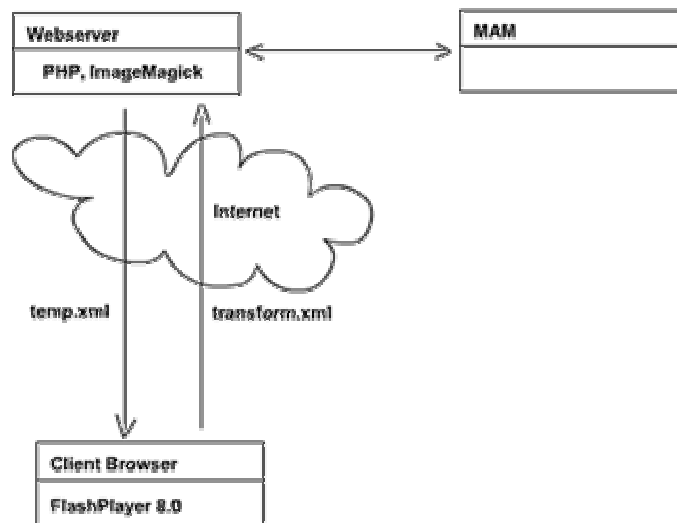
Das Übergebene XML wird folgendes Format beinhalten. Der Temporäre und Originale Pfad wird an das Script zurückübergeben, damit die Temporäre Datei gelöscht und die Originaldatei bearbeitet werden kann.

Transform XML

```
<?xml verion="1.0" encoding = „iso-8859-1“>
<root>
  <tmppath> temppath </tmppath>
  <imgpath> imgpath </imgpath>
  <image_width> px </image_width>
  <image_height> px </image_height>
  <colorspace>rgb</colorspace>
  <transformations>
    <transform action='resize'>
      </param value='width' data='300'>
      </param value='height' data='400'>
    </transform>
    <transform action='rotate'>
      </param value='degrees' data='90'>
    </transform>
    <transform action='changeMode'>
      </param value='mode' data='grayscale'>
    </transform>
    .
    .
  </transformations>
</root>
```

Action	Parameter
rezise	width (int 0..n) height (int 0..n)
scale	value (int 0..200)
crop	x (int 0..n) y (int 0..n) width (int 0..n) height (int 0..n)
rotate	value (0,90,180,270)
brightness	value (-100..100)
contrast	value (-100..100)
saturation	value (-100..100)
hue	value (-100..100)
filter	value("filter_sepia", „filter_blackwhite“, „filter_grayscale“, "filter_blackwhite", "filter_invert")
draw_rectangle	x (int 0..n) y (int 0..n) width (int 0..n) height (int 0..n) fill (color 0x ...) stroke (color 0x ...) stroke-width (int 0-6)
draw_border	stroke (color 0x ...) border-style (int 0 - 6)
draw_line	x1 (int 0..n) x1 (int 0..n) x2 (int 0..n) x2 (int 0..n) stroke (color 0x ...) stroke-width (int 0-6)
draw_text	x (int 0..n) y (int 0..n) text (string) size (int 0..n in Pixel)

Kommunikationsschnittstellen



Die Kommunikation zwischen Client und Server erfolgt über gewöhnliche http-Requests. Der Flashfilm wird aufgerufen und beinhaltet in seinen Parametern die Metadaten des zu bearbeitenden Bildes in XML Format. Das Bild kann bearbeitet werden, Änderungen können persistent auf dem Dateisystem abgelegt werden um eine spätere Bearbeitungen zu ermöglichen. Ein Aufruf auf den Webserver wendet die im XML Format übergebenen Transformationen auf das Originalbild an und stellt das Ergebnis zum Download bereit.

Persistente Speicherung der Transformationen

Der Benutzer soll die Möglichkeit erhalten seine Änderungen an einem Bild persistent zu speichern. Dazu werden die in Flash gemachten Änderungen, wie im ersten Kapitel beschrieben, in das XML Format überführt. Leider bietet Flash nicht die Möglichkeit auf speichernde oder ladende Funktionen des Betriebssystems zuzugreifen, da Flash sonst die Securitysandbox des Flashplayers verlassen würde. Somit bleiben nur 2 Alternativen übrig ein persistentes Speichern zu realisieren.

Shared Objects

Flash kann die gemachten Transformationen in einem SharedObject auf dem Dateisystem speichern. SharedObject sind Cookie ähnliche Dateien, die nur vom FlashPlayer lesbar sind. Des weiteren ist die Größe pro SharedObject auf 100kB begrenzt. Somit kann nur die Möglichkeit angeboten werden das zuletzt bearbeitete Projekt in einem SharedObject zu speichern. (Nicht Realisiert!)

FileUpload / Download

Das erstellte Transform XML wird über die bereits beschriebene Methode „sendAndLoad“ an ein Serverscript geschickt, welches die Datei auf dem Server persistent abspeichert und den Link der Datei an Flash zurückgibt. Flash kann dann ein Download PopUp öffnen, welches die Datei auf das Dateisystem herunterladen lässt.

```
XML.sendAndLoad(url:String, targetXMLObject:XML) : Void  
FileReference.download(url:String, [defaultFileName:String]) : Boolean
```

Equivalent läuft das Laden einer Datei. Flash kann ein Uploadfenster öffnen, welches eine Datei vom Dateisystem auswählen lässt um diese auf den Webserver zu laden. Flash kann nun die Datei vom Webserver laden um die Transformation am Bild zu reproduzieren.

```
FileReference.upload(url:String) : Boolean  
XML.load(url:String) : Boolean
```

Das Sicherheitskonzept von Flash erlaubt nur Uploads in die gleiche Sandbox. Soll die Datei also auf eine andere Domain oder Subdomain geladen werden, müssen die Sicherheitseinstellungen beachtet werden, da Flash hier auf eine andere Sandbox zugreift.

Hier genügt es ein cross-domain policy file (crossdomain.xml) auf der betreffenden Domain abzulegen. Siehe http://www.macromedia.com/go/flash8_security

Multilanguage Support

Die Bildbearbeitung soll eine Mehrsprachfähigkeit unterstützen. Dazu lässt in den Parametereinstellungen von Flash eine Sprache eintragen, die in den Root des Filmes übergeben wird. Dem object-Tag im HTML muß folgende Eigenschaft hinzugefügt werden.

```
<param name="FlashVars" value="language=de">
```

Flash lädt anhand dieses Parameters die Sprachdatei. Die Sprachdatei in dem Verzeichnis „language/<Sprachkürzel>“ abgelegt. Beispielsweise für Deutsch „language/de/“ oder Englisch „language/en“. Sollte dieser Parameter nicht übergeben werden, wird als Standardsprache „de“ ausgewählt.

Das Format der Sprachdatei wird im XLIFF Standard geführt. XLIFF steht für „XML Localization Interchange File Format“. Die Idee – ebenso wie bei OpenTag – besteht darin, dass die vielen unterschiedlich zu lokalisierenden Daten auf ein einheitliches Format abgebildet werden.

Im folgenden wird eine Beispiel XLIFF Datei gezeigt. „editor.xlf“

```
<?xml version="1.0" encoding="UTF-8" ?>  
<xliff version="1.0" xml:lang="de">  
  <file source-language='de' datatype='plaintext'>  
    <header />  
    <body>  
      <group resname="gui">  
        <trans-unit resname="ok">  
          <source>OK</source>  
        </trans-unit>  
        <trans-unit resname="cancel">  
          <source>Abbrechen</source>  
        </trans-unit>  
      </group>  
    </body>  
  </file>  
</xliff>
```

XLIFF 1.1 Specification

<http://www.oasis-open.org/committees/xliff/documents/cs-xliff-core-1.1-20031031.htm>

Actionscript

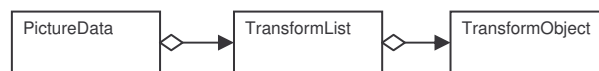
Codeaufbau

Die Codebase liegt im Projektordner im Unterordner „as“ der den Actionscriptteil des Projektes bezeichnet. Um das Projekt in eine Actionscript Package Struktur zu integrieren liegt der Code im Package „de/hdm“. Ab hier wird der Code in den Code des Editors („editor“), einiger Hilfstoools („utils“) und dem Parser der XLIFF Dateien („xliffparser“).

Im Editor werden die Packages nach Daten, GUI, Controller, Transformation und Style getrennt um die Verteilung des Codes der Verwendung anzupassen. Im Folgenden werden die 5 Packages genauer erläutert. Eine genau Codedokumentation ist den Source Files selber zu entnehmen.

Daten (data)

Dieses Package beinhalten alle Objekte die zur Speicherung eines Bearbeitungszustandes notwendig sind.



Desweiteren stellt dieses Package eine Klasse *DataLoader* bereit um Externe Daten zu laden und den Ladefortschritt über Events zurückzugeben. Externe Daten können Bilder, MovieClips oder XML Dateien sein. Die Klasse *GUIState* definiert den Aufbau der GUIs.

GUI (gui)

Das gui Package ist das größte da hier alle Events der Benutzeroberfläche zusammenlaufen verarbeitet und weitergereicht werden.

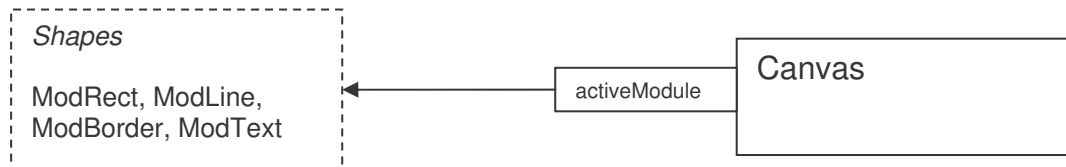
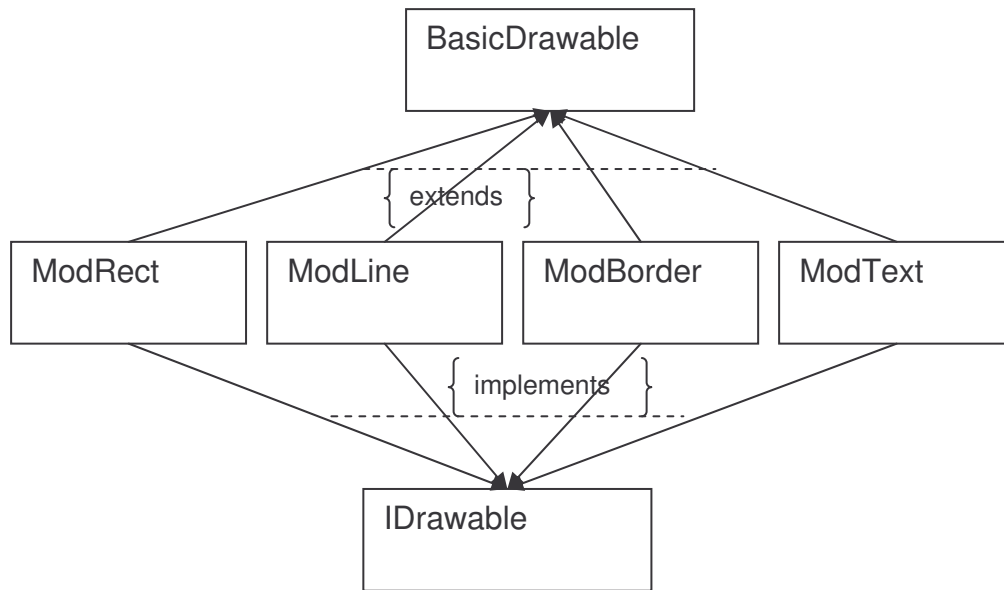
Die Klasse *MainScreen* ist direkt mit dem MovieClip auf der Bühne verknüpft und erbt von der Klasse *Screen* welche Basisfunktionalität zum instanziiieren von weiteren Komponenten bereitstellt. Da hier alle Komponenten der Bühne registriert sind und Events dieser zu dieser Klasse delegiert werden ist der Umfang entsprechend groß. Alle verwendeten Komponenten stammen aus dem Basissatz von Macromedia welches Fenster, Buttons, Slider, Alerts, ComboBoxen, Textfelder, Menüleisten usw. umfasst.

Lediglich der CellRenderer und der ColorChooser sind eigen entwickelte Komponenten. Diese befinden sich im Package „controls“. CellRenderer sind Komponenten die Werte empfangen, darstellen und zurückgeben können um Zellen einer ComboBox, Datagrids usw. zu füllen. In dieser Applikation werden sie für Comboboxen verwendet um die Linienstärke anzuzeigen. Der ColorChooser zeigt eine Farbpalette an und gibt die ausgewählte Farbe zurück.

Das Package „classes“ beinhaltet den Inhalt aller Fenster. Jedes Fenster wird über die Methoden *setPicture* und *modelChange* von Änderungen am Datenmodell berichtet und kann darauf reagieren. Die Methode *setPicture* übergibt zu dem ein *PictureData* Objekt über das sich jedes Fenster beim Datenmodell registrieren kann. Änderungen daran führen also zum Neuzeichnen des GUI, welches sich auch am Modell registriert hat und informiert die anderen Fenster über die registrierte Methode *modelChange* von der Änderung an den Daten.

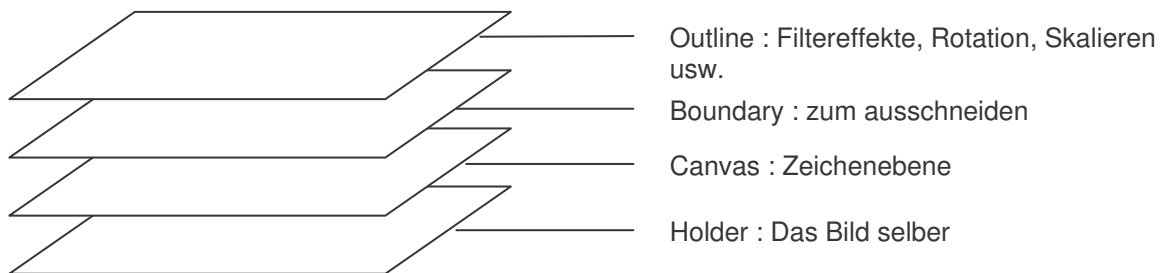
DRAWING

Das Package drawing beinhaltet alle Zeichnungsfunktionen des GUI's und bedient sich bereits vorhandener Klassen um diese zu realisieren. Zeichenmodule lassen sich beliebig erweitern. Wie ein Zeichenmodul sich in das Canvas integriert sollen folgende UML Diagramme verdeutlichen.



EBENEN

Die Zeichenfläche selbst besteht aus mehreren Ebenen die einer Funktion zugewiesen werden. Das Canvas ist eine dieser Ebenen.



Manager (manager)

DISPLAYMANAGER

Der DisplayManager bietet Factory Methoden an um schnell Hinweisfenster, Bestätigungsfenster oder Tooltips anzuzeigen.

TEXTMANAGER

Der TextManager bietet Funktionen an um die geparsete XLIFF Datei auszulesen. Der TextManager wird im Root des Filmes instanziiert um von jeder Position im Programm über `_root.TextManager.getText()` auf die Texte zuzugreifen. Nicht die feinste Variante, da die Typisierung ausgehebelt wird, aber es ist eine einfache Möglichkeit.

Styles (styles)

Hier auf die Funktionsweise von Styles in Flash einzugehen würde den Rahmen sprengen. Die Klasse *Defaults* stellt für Flash das gleiche dar wie CSS für HTML. So können Stylesheets definiert, vererbt, kopiert werden. Die Defaults Klasse bestimmt also das Aussehen des Filmes. Hier bestände auch die Möglichkeit einen CSS Wrapper anzuhängen um jedes beliebige Tool in ein Enterprise Portal zu integrieren.

Filter

Das Package „de.hdm.editor.transform“ bietet einfach Methoden an um Farbmatrizen zu erstellen und zu multiplizieren. Weitere vorgefertigte Matrizen stellen Standardfilter zur Verfügung (Klasse *StandardTransform*).

Die Klasse *ColorMatrix* wurde von Grant Skinner übernommen. Ich habe mich hier dieser Klasse bedient, da sie Farbtransformationen bereits sehr Wahrnehmungsecht bereitstellt.

Einsatz

HTML

Um das Tool im Web nutzen zu können müssen einige Parameter beim Aufruf übergeben werden. Diese Parameter bestimmen den zu ladenden Datensatz und die Sprache.

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swfla
sh.cab#version=8,0,0,0" width="900" height="700" id="editor"
align="middle">
<param name="allowScriptAccess" value="sameDomain" />
<param name="movie" value="editor.swf" />
<param name="quality" value="high" />
<param name="bgcolor" value="#ffffff" />
<param name="flashvars" value="path=sample/editor1.xml&language=de" />
<embed src="editor.swf" flashvars="path=sample/editor1.xml&language=de"
quality="high" bgcolor="#ffffff" width="900" height="700" name="editor"
align="middle" allowScriptAccess="sameDomain" type="application/x-
shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
```

Die hervorgehobenen Parameter sind zwingend wichtig um einen gezielten Einsatz des Editors zu ermöglichen. Fehlen diese Parameter, wird kein Bild geladen und die Standardsprache „Deutsch“ wird ausgewählt.

Das Serverziel zur Rücküberführung der Transformation habe ich aus sicherheitstechnischen Gründen in den Code selber hineingeschrieben. Sollte das Ziel öffentlich lesbar sein könnten Denial of Service Attacken möglich sein. Sollte der Server intelligent genug sein DoS Attacken zu erkennen, kann das Serverziel im Meta Tag des XML Dokuments hineingeschrieben werden.

```
<?xml verion="1.0" encoding = „iso-8859-1“>
<root>
.
.
.
<meta><serverurl>http://www.myserver.de/script.php</serverurl></meta>
</root>
```

Impressum

Autor:	Frank Wenzel
Email:	mail@elunatics.de
Studiengang:	Medieninformatik – Wintersemester 05/06
Betreuender Professor:	Prof.Dr.-Ing. Oliver Kretzschmar
Tools:	Eclipse mit ASDT Plugin und Macromedia Flash 8
Externe Bibliotheken:	Ladizlav Zigo (Tweening Package) Grant Skinner (ColorMatrix) Marc Bauer (DrawingUtils)

Nachwort

Das Tool hat noch nicht den fertigen Endstatus erreicht und wird von mir privat weiterentwickelt.

Geplante Tasks sind:

- Undo Funktion mit Memento Pattern um auch ein Redo zu ermöglichen
- Erweiterung der Zeichenpalette
- Entwickeln eines PHP Scripts zur Bearbeitung des Originalbildes
- Erweitern der Filter, Speichern eigen erstellter Filter
- Bugs entfernen ... :-/

Der Fortschritt wird auf www.elunatics.de veröffentlicht, sobald meine Seite wieder online ist.