

Institut für Wissensmedien

## Studienarbeit

Entwicklung interaktiver Quizzes mit Hilfe von Adobe  
Flash, MTASC und XML

Thomas Lempa

Koblenz, den 27. Juli 2006

**Prüfer:** Dr. Ingo Dahn

**Betreuer:** Dipl.-Inf., Dipl.-Ing.(FH) Marc Santos

## Selbstständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Ja    Nein

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

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.          
Koblenz, den 27. Juli 2006

---

Lempa Thomas

## Abstract

**Zielsetzung:** Für die Feuerwehr- und Katastrophenschutzschule Rheinland - Pfalz (LFKS) sollen auf der Basis von Flash und XML interaktive Quizzes realisiert werden, die im Rahmen des Gruppenführerlehrgangs als Lernerfolgskontrolle dienen sollen. Als Grundlage wird ein selbstentwickeltes MVC Framework und Adobe (ehemals Macromedia) Flash herangezogen.

**Entwicklung:** Bei der Entwicklung der Quizzes wurde sehr viel Wert auf eine systemunabhängige, bestmöglich standardisierte Datenhaltung gelegt. Hier war die Verbindung von Flash und XML eine optimale Lösung.

**Technik:** Mit der Kombination aus ActionScript2, XML und Flash ist es möglich, innerhalb kürzester Zeit unterschiedliche Quizzes zu erstellen. Der Grundgedanke bei dieser Technik ist es, Daten und ihre Repräsentation zu trennen. Die Flash Entwicklungsumgebung wird für die Erstellung der Views (Darstellungselemente) verwendet. Die Views werden anhand von dynamischen Textfeldern, interaktiven Animationen und weiteren Darstellungselementen definiert. Diese Views werden über das MVC-Framework mit Actionscript2 angesprochen und mit Inhalten aus einer externen XML Datei gefüllt. Der Vorteil dabei ist, dass die Inhalte und zum Teil auch das Design recht schnell ausgetauscht werden können, ohne an der Applikation Änderungen vornehmen zu müssen. Die XML Dateien werden mit Hilfe der Software Infopath der Firma Microsoft erstellt. Eine weitere Möglichkeit die Daten zu standardisieren ist das IMS Question and Test Interoperability (IMS QTI) Datenformat, welches explizit für solche Anwendung entwickelt wurde. Dieses Format wird jedoch nur am Rande behandelt.

Diese Flashapplikationen können sowohl als eigenständige Programme betrachtet oder als Elemente einer HTML Datei in einem Browser dargestellt werden. Aus diesem Grund ist es möglich, auf fast jedem Gerät (PC, Notebook, Handheld und Mobiltelefon) und auf jedem Betriebssystem (Windows, Mac OS, Linux, etc.) eine Flashapplikation zum Laufen zu bringen.

**Ausblick:** Durch das selbstentwickelte MVC-Framework kann die Applikation schnell und einfach erweitert werden. Es wurde jedoch drauf geachtet, dass die Applikation auch ohne den Quellcode neu zu kompilieren vom Aussehen verändert werden kann.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>iii</b>
<b>Listings</b>	<b>iv</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Gliederung der Arbeit . . . . .	1
1.3 Feuerwehr- und Katastrophenschutzschule Rheinland-Pfalz . . . . .	2
1.4 Ziel . . . . .	2
<b>2 eLearning</b>	<b>3</b>
2.1 Definition . . . . .	3
2.2 Vorteile . . . . .	3
2.3 Technologien . . . . .	4
2.3.1 Web- und Computer-basierte Trainingsanwendungen . . . . .	4
2.3.2 Autorensysteme . . . . .	4
2.3.3 Learning Management Systeme . . . . .	5
2.3.4 Learning Content Management Systeme . . . . .	5
<b>3 Implementierung</b>	<b>6</b>
3.1 Einleitung . . . . .	6
3.2 Flash . . . . .	6
3.3 MTASC . . . . .	7
3.4 Actionscript . . . . .	7
3.5 OOP . . . . .	8
3.6 XML . . . . .	8
3.6.1 Infopath . . . . .	11
3.7 MVC . . . . .	12
3.8 MVC Framework . . . . .	13
3.9 Quizaufbau . . . . .	16
3.9.1 Die Präsentationsschicht (View) . . . . .	16
3.9.2 Die Programmschicht (Controller) . . . . .	18
3.9.3 Das Datenmodell (Model) . . . . .	20
3.9.4 Softwaretechnische Sicht . . . . .	20

<b>4</b>	<b>Ausblick</b>	<b>22</b>
4.1	Standards . . . . .	22
4.1.1	IMS QTI . . . . .	23
<b>5</b>	<b>Zusammenfassung</b>	<b>24</b>
	<b>Literaturverzeichnis</b>	<b>25</b>

## Abbildungsverzeichnis

3.1	Infopath Drag and Drop . . . . .	11
3.2	Infopath Formular zum Ausfüllen . . . . .	12
3.3	Framework Ordnerstruktur . . . . .	13
3.4	Framework Ordner- und Dateistruktur . . . . .	14
3.5	Übersicht Bibliothek . . . . .	17
3.6	Freigabe für Actionscript . . . . .	18
3.7	Klassendiagramm . . . . .	21

## Listings

3.1	Hello World in Actionscript . . . . .	8
3.2	MTASC Aufruf zur Kompilierung . . . . .	8
3.3	XML Beispieldatei . . . . .	9
3.4	XML Eigenschaften . . . . .	10
3.5	Makefile . . . . .	15
4.1	Beispiel swfmill XML Datei . . . . .	22

# 1 Einleitung

Für die Feuerwehr- und Katastrophenschutzschule Rheinland – Pfalz (LFKS) sollen auf der Basis von Flash und XML interaktive Quizzes realisiert werden, die im Rahmen des Gruppenführerlehrgangs als Lernerfolgskontrolle dienen sollen. Im Rahmen der Studienarbeit wird mit Hilfe von Adobe Flash und einem selbst entwickelten Framework ein Quiz entwickelt.

Das Augenmerk richtet sich hier sowohl auf das Multiple-Choice-Verfahren als auch auf das Drag and Drop Verfahren für Textfragen und Bilder. Die Auswahl der Art, wie auch weitere Gestaltungseigenschaften werden in der XML Datei definiert.

## 1.1 Motivation

eLearning ist eine moderne Methodik der Wissensvermittlung, welche immer mehr an Bedeutung gewinnt. Immer mehr Bildungseinrichtungen und Hochschulen setzen eLearning-Plattformen im Rahmen der Bildung ein. Geschätzte 3-5% [Dah] der Studenten an den deutschen Hochschulen werden in die eLearning-Programme einbezogen und Jahr für Jahr werden es immer mehr.

Das Institut für Wissensmedien der Uni Koblenz konzipiert, implementiert und evaluiert einen Blended Learning-Lehrgang zur Ausbildung von Gruppenführern der Freiwilligen Feuerwehr an der Feuerwehr- und Katastrophenschutzschule Rheinland-Pfalz (LFKS). Das IWM setzt hierfür die eLearning Plattform WebCT ein.

## 1.2 Gliederung der Arbeit

Voraussetzung für diese Studienarbeit sind mit Infopath entwickelte XML Dateien. Diese Datei beinhaltet alle Formatierungen und den Inhalt für das Quiz. In Kapitel 2 wird auf die Quiz-Komponenten der unterschiedlichen eLearning Systeme eingegangen. Hierbei liegt der Schwerpunkt auf dem eLearning System WebCT. Im folgenden Kapitel wird die verwendete Technik aufgeführt und der Aufbau aus softwaretechnischer Sicht dargestellt. Im letzten Kapitel wird auf die weiteren Erweiterungsmöglichkeiten und XML Standards eingegangen.



## 1.3 Feuerwehr- und Katastrophenschutzschule Rheinland-Pfalz

Die Feuerwehr- und Katastrophenschutzschule (LFKS) ist die zentrale Ausbildungsstätte für die Freiwilligen Feuerwehren, die Berufs- und Werkfeuerwehren und die anderen Hilfsorganisationen in Rheinland-Pfalz. In rund 250 Veranstaltungen werden jährlich ca. 5.300 Kräfte der verschiedenen Organisationen geschult.

Damit hat sich sowohl die Anzahl der Lehrgangsteilnehmer als auch die der Lehrgangstage in den letzten zehn Jahren mehr als verdoppelt. Die Anzahl der Lehrgangsteilnehmer aus höher qualifizierten Berufen ist dem gegenüber stetig gesunken.

Die Gründe dafür liegen hauptsächlich in der Pflicht der Arbeitgeber, diese Mitarbeiter für die Lehrgänge und Feuerwehreinsätze von der Arbeit freizustellen.

Um dieser Tendenz entgegen zu wirken, hat sich die Leitung der LFKS über neue Lehr- und Lernmethoden informiert und intensiv mit den Themen E-Learning und Blended Learning auseinander gesetzt.

Nach zahlreichen Gesprächen mit Firmen und Institutionen wurde schließlich dem Institut für Wissensmedien (IWM) der Universität Koblenz-Landau der Projektauftrag zur Unterstützung der Einführung eines Blended Learning Lehrgangs für die Ausbildung der Gruppenführer der Freiwilligen Feuerwehren erteilt. [IfW]

## 1.4 Ziel

Das Ziel dieser Studienarbeit ist ein dynamisches und einfaches Quiz zu erstellen, welches sowohl für das Projekt Feuerwehr- und Katastrophenschutzschule Rheinland – Pfalz (LFKS) als auch für andere Institutionen verwendet werden kann.

## 2 eLearning

Unter eLearning (englisch electronic learning – elektronisch unterstütztes Lernen), auch E-Lernen genannt, werden alle Formen von Lernen verstanden, bei denen digitale Medien für die Distribution von Lernmaterialien und/oder die Kommunikation zum Einsatz kommen. [Wika]

### 2.1 Definition

eLearning ist ein indirekter Lernprozess (Übermittlung von Wissen und Informationen) mit Hilfe der modernen Informationstechnologie. Diese Technologie wird für die Erschaffung, Verarbeitung, das Senden und die Präsentation von Inhalten und Informationen, Lehre und Wissen verwendet. eLearning ist mit der Idee der ständigen Entwicklung und Erhöhung der Qualifikation mit bestmöglichen Methoden verbunden.

Allerdings ist die Technologie bei eLearning nicht alles. eLearning verbindet die Problematik auf sowohl die technische Frage „Auf welche Art und Weise kann ein Lehrgang vorbereitet und durchgeführt werden?“ als auch auf die humanwissenschaftliche Frage „Wie kann man erfolgreich schulen?“ zu antworten. In diesem Sinn hat eLearning einen interdisziplinären Charakter.

Die Idee von eLearning ist nicht die traditionellen Lernmethoden zu verdrängen, sondern diese zu vertiefen und zu ergänzen. Die Wahl der gemischten Lernform erlaubt es, die besten Merkmale des Direkt- und Fernunterrichts bei gleichzeitiger Kostenreduktion und Erhöhung der Attraktivität didaktischer Inhalte auszunutzen. Eben das ist eine Möglichkeit, erhebliche Ersparnisse bei den Ausbildungskosten zu bekommen, ohne der Notwendigkeit gleichzeitig auf die Qualität des Lernprozesses zu verzichten. Dies ist die Motivation für die Popularität und schnelle Weiterentwicklung von eLearning. [Wika]

### 2.2 Vorteile

Vorteile des eLearnings im Gegensatz zum traditionellen Lernens kann man wie folgt klassifizieren:

- niedrige Kosten und höhere Effizienz,
- einfache Verwaltung didaktischer Inhalte,
- elastisches und schnelles fungieren,

- individueller Lernprozess,
- mehrmaliges Ableisten der Lerninhalte,
- eigenständige Wahl der Reihenfolge und des zu lernenden Wissensgebiets,
- ausführliche Statistiken und genaue Beurteilung des Lernfortschritts durch den Lehrer,
- Tempo des Lernens selbst bestimmbar.

[Aka]

## 2.3 Technologien

Die Lernprozesse können durch unterschiedliche Technologien sowie unterschiedliche didaktische Szenarien realisiert werden. Man unterscheidet diese Plattformen in folgenden Varianten

- Web- und Computer-basierte Trainingsanwendungen
- Autorensysteme
- Learning Management Systeme
- Learning Content Management Systeme

### 2.3.1 Web- und Computer-basierte Trainingsanwendungen

Das sogenannte Computer Based Training (CBT) ist ein Lernprogramm, welches von den Lernenden unabhängig von den Lehrenden und anderen Lernenden benutzt werden kann. Diese Form von Training wird prinzipiell auf CD-Roms und DVDs mit multimedialen Inhalt vertrieben. Eine Erweiterung des CBT ist das Web Based Training (WBT). Hierfür werden die Lern- einheiten nicht auf Datenträger sondern über das Internet vertrieben. Durch diese multimediale Form besteht die Möglichkeit weitere interaktive Elemente, wie ein Chat, News oder Video- und Audiostreamings, einzubinden.

### 2.3.2 Autorensysteme

Im Gegensatz zu anderen Systemen bieten die Autorensysteme eine visuelle Programmierumgebung, eine sogenannte IDE. Diese Umgebung verhilft den Autoren bei der Erstellung von digitalen Lernangeboten, ohne ein größeres Wissen über die Programmiersprachen oder einer Auszeichnungssprache wie XML , HTML oder ähnlichen zu

haben. Die Lerninhalte werden meist auf CD-Rom aber auch über das Internet vertrieben. Der Schwerpunkt bei den Autorensystemen liegt darauf, dass die Lerninhalte nur für einen bestimmten Zweck entwickelt werden, so dass diese nicht als Modul für andere Lerninhalte verwendet werden können. Es gibt jedoch leistungsfähigere Autorensysteme, welche modular aufgebaut sind und daher mit Hilfe von XML in anderen Systemen bzw. Lerninhalten verwendet werden können.

### **2.3.3 Learning Management Systeme**

Ein Learning Management System, kurz LMS, ist ein System mit welchem man in Autorensystemen erstellte Lerninhalte verwalten kann. Diese Lerninhalte werden in einer Datenbank gespeichert und können somit mehrfach verwendet werden. Es werden jedoch nicht nur die Lerninhalte verwaltet, sondern auch weitere Aufgaben wie z.B. Planung, Anmeldung, Bereitstellung der Kursunterlagen, Zertifizierung und die Erfolgsmessung. Somit kann mittels des LMS der ganze Kursplan durch den Tutor verwaltet werden. Zusätzlich können die Lehrenden und die Lernenden mit Hilfe dieser Plattform kommunizieren.

Bei der Kommunikation gibt es 2 unterschiedliche Wege; die asynchrone Kommunikation, welche z.B. Email, Foren und Dokumentenaustausch beinhaltet und die synchrone, welche z.B. Chats und Writeboards umfasst.

### **2.3.4 Learning Content Management Systeme**

Ein Learning Content Management System (LCMS) ist eine Erweiterung des LMS. Die Aufgabe des LCMS ist die Verwaltung des Contents der Lernobjekte. Im Gegensatz zum LMS werden diese nicht aus anderen Autorensystemen importiert, sondern direkt in dem System erstellt. Auf diese Weise können sie verändert und verwaltet werden. Der Vorteil dabei ist, dass es für den Inhalt eine Versionskontrolle gibt, welche die Möglichkeit bietet, immer wieder auf ältere Inhalte zurückgreifen zu können.

Eine Multi-User-Funktionalität erlaubt unterschiedlichen Personen verschiedene Zugriffsrechte zu vergeben und die Gefahr, dass mehrere Personen am gleichen Inhalt arbeiten, wird dadurch vermieden.

## 3 Implementierung

In diesem Kapitel wird der Aufbau und die Funktionsweise des Quizzes beschrieben und alle verwendeten Programme für die Implementierung dargestellt.

Bei Beginn der Studienarbeit war das Ziel, ein Quiz mit Hilfe von Adobe Flash eLearning Komponenten zu erstellen. Leider sind diese Komponenten sehr kompliziert geschrieben und in Flash integriert, so dass es nicht effizient gewesen wäre, diese Komponenten zu erweitern, damit die Inhalte über eine XML Datei eingelesen werden könnten. Aus diesem Grund wurde ein eigenes Framework entwickelt, welches mit Hilfe des Open-Source Compilers MTASC (3.3) die Möglichkeit bietet, einfacher eine Anwendung zu implementieren.

### 3.1 Einleitung

Für die Entwicklung des Quizzes wurden Adobe Flash für die Visualisierung, MTASC für die Kompilierung und das entwickelte Framework eingesetzt. Als Schnittstelle für den Inhalt diente XML, welches mit Microsoft Infopath erzeugt wurde.

### 3.2 Flash

Flash ist eine Entwicklungsumgebung zur Erzeugung von Flashfilmen im SWF-Format; es basiert auf Vektorgrafik, bietet jedoch auch die Möglichkeit Bilder und Filme zu integrieren. Die größten Vorteile bei Flash sind die Skalierbarkeit ohne Qualitätsverlust und der geringe Speicherbedarf, was sich in einer geringeren Übertragungszeit widerspiegelt. Die Flashfilme finden im Internet in unterschiedlicher Art und Weise Einsatz; sei es als ein kleines Werbebanner, eine komplette, interaktive Webseite oder ein kleines Spiel. Die Verbreitung der Flashseiten begann mit der Einführung von Actionscript (Siehe 3.4) in der Flash 4 Version.

Durch die Kombination von Animation und Programmierung findet Flash immer mehr Anhänger, da komplexe Webseiten und Programme, wie Online-Spiele und Präsentationen leicht erstellt werden können. Der Vorteil dabei ist, dass Flash inzwischen mit allen möglichen serverseitigen Programmiersprachen kommunizieren und dynamisch erzeugte Inhalte über XML- und Textdateien auslesen kann.

Außerdem ist es mit der Hilfe eines Browserplugins (Flash Player) möglich, eine Flashanimation in jedem Browser und somit auf jeder Plattform (PC, Handheld, Telefon) abzuspielen.

### 3.3 MTASC

Mit MTASC ist es möglich, in einer beliebigen Entwicklungsumgebung geschriebenen Actionscript2 Code ohne Einsatz der Adobe Flash Authoringsuite in eine lauffähige Anwendung zu übersetzen. Den Compiler gibt es als fertig kompilierte Pakete für Windows, Linux und Mac OS X; von der Projektseite<sup>1</sup> kann auch der Quellcode heruntergeladen werden.

Bei der Entwicklung des Quizzes wurde jedoch nicht auf Flash verzichtet. Komplexe Grafiken wurden in Flash aufbereitet und für Actionscript exportiert, um mit MTASC darauf zugreifen zu können. In dem Abschnitt 3.9 wird der genaue Einsatz von Flash beschrieben.

### 3.4 Actionscript

Actionscript wurde 1992 zur Einführung von Flash4 vorgestellt. Hierbei handelt es sich um eine Javascript ähnlichen programminterne Sprache, welche bei der Abspielzeit eines Flashfilmes interpretiert wird. 2003 wurde mit Flash MX 2004 (Version 7) das objektbasierte Actionscript durch die objektorientierte javaähnliche Programmiersprache Actionscript2 abgelöst.

Die Vorzüge der Objektorientierten Programmierung kamen in dieser Version zum Vorschein, werden jedoch noch nicht komplett unterstützt. Die Mehrfach-Vererbung oder die Bildung von abstrakten Klassen ist in ActionScript2 nicht möglich. Der Vorteil dabei ist jedoch, dass externe Actionscript Dateien von Flash eingelesen und interpretiert werden können. Auch durch die Weiterentwicklung der Flash-API (Programmierschnittstelle) eröffnet sich die Möglichkeit, auf die internen Funktionsweisen von Flash zu zugreifen.

Das in der Studienarbeit entwickelte Quiz wurde mit Actionscript2 umgesetzt und nicht mit der damals unausgereiften Version 3 der Sprache.

Ende 2005 wurde die Version 3 von Actionscript vorgestellt. Zusammen mit Actionscript3 wurde auch die Entwicklung der AVM2 (ActionScript Virtual Machine) bekannt gegeben. Die AVM verwendet einen Laufzeit-Compiler, der den ActionScript-Code in nativen Maschinencode übersetzt und damit erheblich beschleunigt. Für die Entwicklung von Webseiten und so genannten Rich Internet Applications (RIA) gibt, wird nicht mehr eine IDE verwendet, sondern das neu entwickelte Authorensystem Flex Builder 2, welches auf dem Open Source Tool Eclipse basiert. Dieses System ist jedoch nur für Programmierer und nicht für Designer gedacht. Die dazugehörige IDE soll im Jahr 2007 vorgestellt werden.

---

<sup>1</sup><http://www.mtasc.org>

## 3.5 OOP

OOP ist für die Erstellung von komplexen Applikationen ausgelegt. Mit dieser Art zu programmieren wird eine Applikation überschaubar und modular zu realisieren. Somit ist es möglich, Teile des Codes für andere Applikationen zu verwenden und durch die Anlehnung an die Programmiersprache Java, ist eine Einarbeitung für andere Entwickler unproblematisch.

Weitere Eigenschaften und Funktionen einer OOP können bei der Vorlesung Informatik A und Informatik B im Studium erlernt werden.

Wie in jeder Programmiersprache beginnt man bei der Einarbeitung mit dem einfachsten Programm und zwar mit „Hello World“ (siehe 3.1).

Listing 3.1: Hello World in Actionscript

```
1 class Hello {
2     function Hello (path :MovieClip) {
3         path.createTextField("tf", 1, 0, 0, 100, 20);
4         path.tf.border = true;
5         path.tf.text = "Hello_World";
6     }
7
8     static function main() {
9         var t :Hello = new Hello(_root);
10    }
11 }
```

Dieser Programmcode wird in einer Actionscript Datei abgespeichert (z.B. Hello.as). Um eine SWF Datei daraus zu generieren, wird die Datei mit Hilfe von MTASC (siehe 3.3) mit folgenden Befehl kompiliert:

Listing 3.2: MTASC Aufruf zur Kompilierung

```
1 mtasc -swf hello.swf -main Hello.as -header 200:200:30
```

Mit diesem Aufruf entsteht die Datei „hello.swf“ welche in einem Browser aufgerufen werden kann.

## 3.6 XML

XMI (Extensible Markup Language) ist eine einfache und flexible Meta- von SGML (Standard Generalized Markup Language) abgeleitete Sprache für Dokumente und beinhaltet strukturierte Informationen.

Der Grundgedanke für den Einsatz von XML ist, Inhalte automatisch zugänglich, auffindbar und editierbar zu machen. In der XML Datei werden Inhalte über kennzeichnende Markierungen, die sogenannten Tags, in funktionelle Blöcke untergliedert. Im Gegensatz zu HTML, welche auch eine Auszeichnungssprache ist, können die Tags beliebig gewählt werden. Meistens werden Namen gewählt, welche die Bedeutung des Inhaltes ausdrücken (z.B. <question>).

Der logische Aufbau eines XML-Dokumentes ist ein hierarchisch strukturierter Baum und beinhaltet folgende Knoten:

- Elemente, welche entweder aus einem leeren Tag (<question />) oder einer Kombination aus einem Start-Tag (<question>) und einem End-Tag (</question>) bestehen
- Attribute (attributname=„attributwert“), welche Zusatzinformationen der Elemente beinhalten
- Verarbeitungsanweisungen (<?Name Daten?>)
- Kommentare (<!-- Kommentar-Text -->)

Das XML Dokument muss mit einem Wurzelknoten beginnen und das Innere des Dokuments vollständig einschließen.

Die in der Studienarbeit verwendete XML Datei (siehe 3.4) wurde zuerst in einem Texteditor erstellt.

Listing 3.3: XML Beispieldatei

```
1 <quiz bgcolor="0xFFFFFFFF" selection="0" bgrade="0x950C06"  
2 fontc="0xFFFFFFFF" logoURL="images/logo.jpg"  
3 bggrads="0x40A6DC" bggrade="0xFFFFFFFF" bcolors="0xEB130A"  
4 description="Das_ist_eine_Beschreibung" extrac="0xFF0000"  
5 title="Multiple_Choice_Quiz" hcolor="0x950C06"  
6 bgrads="0xEB130A" bcolore="0x950C06" ecolor="0xFFFFFFFF"  
7 errorc="0x000000">  
8     <question text="" answer="" imageURL="">  
9         <selection ></selection >  
10    </question >  
11 </quiz >
```

In dem Wurzelknoten der XML Datei werden viele Attribute (siehe ??) definiert, welche das Aussehen und die Funktionsweise des Quizzes darstellen.



Listing 3.4: XML Eigenschaften

```
1 Allgemeine Attribute der Quiz Node:
2 title           = Titel des Quizzes
3 description     = Beschreibung des Quizzes
4 selection       = Art des Quizzes
5     0 = Multiplechoice
6     1 = Drag and Drop
7     2 = Multiplechoice fuer Bilder
8     3 = Drag and Drop fuer Bilder
9 logoURL         = Pfad zum Logo
10
11 Farbattribute (Die Farbwerte sollen im Hexcode):
12 bgcolor         = Farbe des Hintergrunds
13 bggrads        = Startfarbe fuer den Hintergrundverlauf
14 bggrade        = Endfarbe fuer den Hintergrundverlauf
15 hcolor         = Farbe fuer den Kopfbereich
16 bgrads        = Startfarbe fuer den Barrenverlauf
17 bgrade        = Endfarbe fuer den Barrenverlauf
18 bcolors        = Startfarbe fuer den Buttonverlauf
19 bcolore        = Endfarbe fuer den Buttonverlauf
20 ecolor         = Farbe fuer Kanten und Linien
21 fontc         = Allgemeine Schriftfarbe
22 errorc         = Besondere Schriftfarbe (z.B. Feedbackausgabe)
23 extrac        = Extra Farbe (Farbe der Auswahlkasten)
24
25 Attribute der Question Node:
26 text           = Frage (ohne Fragezeichen)
27 answer         = Die richtige Antwort
28 imageURL (optional) = Bildpfad zu der aktuellen Frage
29                                     (Bilder sollen quadratisch sein)
30
31 Elemente der Selection Node:
32 Zwischen jedes Element kommt eine Antwort.
33 Bei Bildquiz sollen die Bildnamen eingetragen werden
```

Die mit der Hand erstellte XML Datei ist für einen Tutor recht schwierig zu pflegen und zu verwalten. Um diesen Prozess zu vereinfachen, wurde ein Microsoft Infopath Template erstellt.

### 3.6.1 Infopath

Microsoft Infopath bietet die Möglichkeit, Formulare zu gestalten, welche im Nachhinein ausgefüllt werden können. Infopath kann in Verbindung mit Microsoft SharePoint Server genutzt werden, um die Formulare zu verwalten und sie anderen Anwendungen zur Verfügung zu stellen.

Das Erstellen eines Infopath Templates erfordert eine kurze Einarbeitung in das Programm. Nachdem man sich in die Beispiel-Formulare eingelesen hat, kann man schnell ein eigenes Formular entwickeln. Die am Anfang erstellte XML Datei kann in ein leeres Infopath Entwurfsdokument als Datenquelle importiert werden; eine Erstellung eines Templates ohne Datenquelle ist jedoch auch möglich. Durch einfaches Drag and Drop (Bild 3.1) können die Elemente innerhalb des Formulars beliebig positioniert und beschriftet werden.

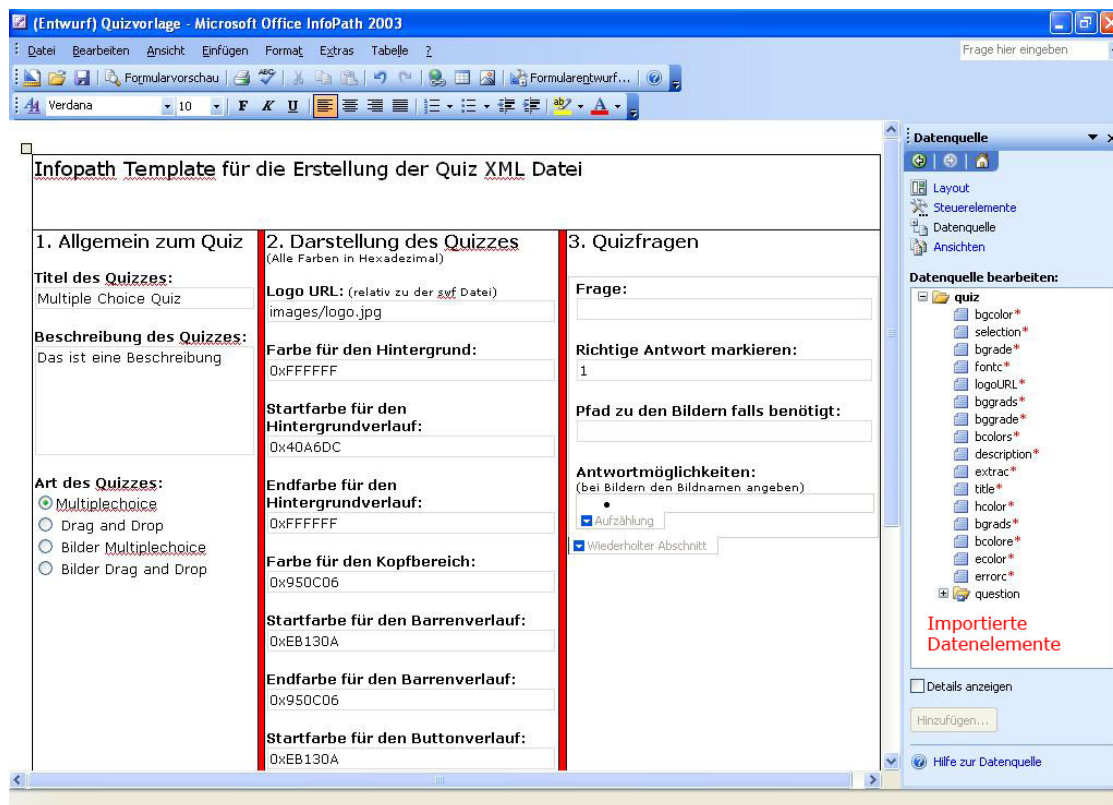


Abbildung 3.1: Infopath Drag and Drop

Nachdem alle Datenelemente integriert worden sind und das Layout des Formulars angepasst wurde, wird das Formular als ein Template gespeichert.

Beim nächsten Aufruf des Templates wird das Formular nicht als Entwurf sondern als Eingabeformular (Bild 3.2) geöffnet. Alle Positionen haben einen Standardwert, können jedoch beliebig geändert werden.

Infopath Template für die Erstellung der Quiz XML Datei

1. Allgemein zum Quiz	2. Darstellung des Quizzes (Alle Farben in Hexadezimal)	3. Quizfragen
<b>Titel des Quizzes:</b> Multiple Choice Quiz	<b>Logo URL:</b> (relativ zu der swf Datei) images/logo.jpg	<b>Frage:</b> _____
<b>Beschreibung des Quizzes:</b> Das ist eine Beschreibung	<b>Farbe für den Hintergrund:</b> 0xFFFFFFFF	<b>Richtige Antwort markieren:</b> 1
<b>Art des Quizzes:</b> <input checked="" type="radio"/> Multiplechoice <input type="radio"/> Drag and Drop <input type="radio"/> Bilder Multiplechoice <input type="radio"/> Bilder Drag and Drop	<b>Startfarbe für den Hintergrundverlauf:</b> 0x40A6DC	<b>Pfad zu den Bildern falls benötigt:</b> _____
	<b>Endfarbe für den Hintergrundverlauf:</b> 0xFFFFFFFF	<b>Antwortmöglichkeiten:</b> (bei Bildern den Bildnamen angeben) •
	<b>Farbe für den Kopfbereich:</b> 0x950C06	
	<b>Startfarbe für den Barrenverlauf:</b> 0xEB130A	
	<b>Endfarbe für den Barrenverlauf:</b> 0x950C06	
	<b>Startfarbe für den Buttonverlauf:</b> 0xEB130A	

Speicherort der Formularvorlage: C:\Dokumente und Einstellungen\thomas\Desktop\Quizvorlage.xsn

Abbildung 3.2: Infopath Formular zum Ausfüllen

Das Formular wurde in 3 Teile aufgeteilt. In der linken Spalte werden allgemeine Informationen zum Quiz und die Art des Quizzes eingegeben. Die mittlere Spalte bestimmt das Aussehen des Quizzes und in der rechten Spalte können die Fragen eingegeben werden. Mittels eines Pfeils in der letzten Spalte können einfach weitere Fragen hinzugefügt werden.

Nachdem das Formular komplett ausgefüllt wurde, kann dieses als XML Datei exportiert werden.

### 3.7 MVC

MVC ist die Abkürzung für das Model-View-Controller Paradigma. Das MVC ist ein Architekturmuster zur Aufteilung von Softwaresystemen in drei Einheiten:

- Das Datenmodell (*engl. Model*) repräsentiert die Daten einer Anwendung
- Die Präsentation (*engl. View*) übernimmt die Darstellung der Daten
- Die Programmsteuerung (*engl. Controller*) definiert, wie ein Benutzer mit der Anwendung interagieren kann

Der Grundgedanke bei MVC ist die Trennung der Präsentation, der Daten und der Interaktion. Durch diese Teilung der Objekte ist die Realisierung der gleichen Anwendung für unterschiedliche Endgeräte möglich. Es wird zuerst die Datenlogik (Model) erstellt, welche an unterschiedliche Benutzeroberflächen (View) gebunden werden kann. Der Controller stellt sowohl die Transformation und die Programmlogik (businesslogic) bereit, als auch die Kommunikation zwischen Model und View her. Desweiteren wird der programmierte Code übersichtlicher und modularer, so dass einzelne Bereiche der Anwendung wiederverwendet werden können und einzelne Elemente leicht ausgetauscht werden können.

## 3.8 MVC Framework

Flash ist eine Entwicklungsumgebung, mit welcher man keine Applikation in dem MVC Paradigma entwickeln kann. Aus diesem Grund wurde ein Framework erstellt, welches auf dieses Paradigma zurückgreift, und mit welchem man Flashapplikationen erstellen kann. Der Aufbau des Frameworks wurde dem Aufbau des Open Source Frameworks Ruby on Rails<sup>2</sup> angelehnt. Um die Übersichtlichkeit im Sourcecode zu behalten, wurden die Dateien in einer bestimmten Ordnerstruktur untergebracht (vgl. 3.3).

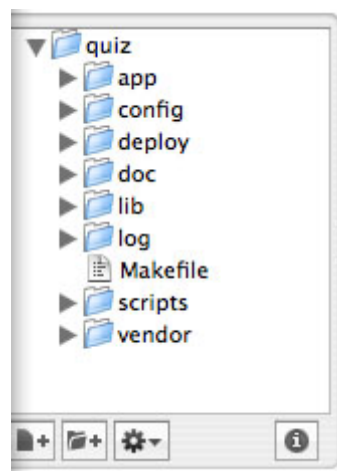


Abbildung 3.3: Framework Ordnerstruktur

Das komplette Framework besteht aus 8 Ordnern und mehreren Unterordnern (vgl. 3.4). In dem Ordner „app“ befindet sich die Struktur der entwickelten Anwendung. Dieser besteht aus der typischen MVC Struktur mit den Ordner „models“ , „views“ , „controllers“ und einem zusätzlichen „helpers“ Verzeichnis mit kleinen Helferklassen. Der „config“ Ordner beinhaltet die Umgebungsdatei „environment.as“ ,in welcher sich die Startfunktion „main“ befindet und weitere Flasheigenschaften definiert werden. Die

---

<sup>2</sup><http://www.rubyonrails.org/>

Ordner „lib/apis“ und „lib/apps“ enthalten die verwendeten MTASC und weitere Open Source Actionscript Klassen. Zusätzlich befindet sich im „apps“ Ordner der MTASC Kompiler für alle gängigen Betriebssysteme. Der Ordner „core“ besteht aus weiteren Hauptbestandteilen des Frameworks. In dem Ordner „scripts“ befindet sich ein separater Flash Player und eine in Flash geschriebene Debugger Console. Der letzte Ordner beinhaltet alle Actionscript Klassen, welche von Flash selbst verwendet werden. Dadurch, dass alle Actionscript Klassen und der Kompiler MTASC im Framework integriert sind, ist es einem Programmierer möglich, ohne weitere Installation von Flash oder MTASC das Framework zu benutzen.

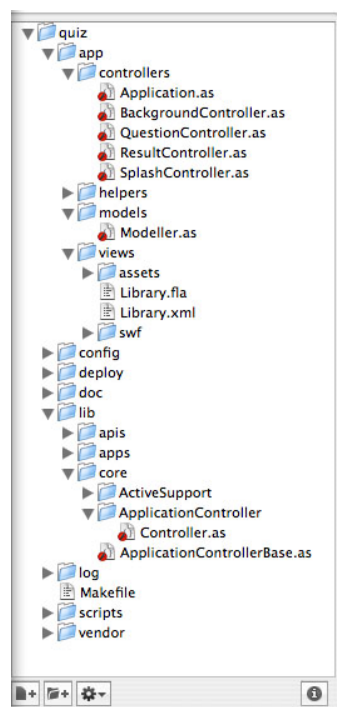


Abbildung 3.4: Framework Ordner- und Dateistruktur

Zuallerletzt befindet sich dort noch ein „Makefile“ (vgl. 3.5), welche das Kompilieren etwas vereinfacht. Die Pfade in der Datei wurden dem Betriebssystem Mac OS X angepasst, so dass man bei anderen Betriebssystemen die Pfade entsprechend anpassen muss.

Listing 3.5: Makefile

```
1 PROJECT = quiz
2 VERSION = 7
3
4 # Applications
5
6 # Mac environment
7 MTASC = ./lib/apps/mtasc
8
9 # Linux environment
10 # MTASC = ./lib/apps/mtasc-linux
11
12 # Windows environment
13 # MTASC = C:\mein Pfad zu\mtasc.exe
14
15 ##### STOP EDITING #####
16
17 CLASSPATH = -cp vendor/ \
18             -cp lib/apis/ \
19             -cp lib/core/ \
20             -cp app/controllers/ \
21             -cp app/helpers/ \
22             -cp app/models/
23
24 sources = $(wildcard app/controllers/*.as) \
25           $(wildcard app/helpers/*.as) \
26           $(wildcard app/models/*.as)
27 library = app/views/swf/Library.swf
28
29 .PHONY: all
30 all: development
31 development: $(sources)
32             $(MTASC) $(CLASSPATH) \
33             -version $(VERSION) \
34             -out deploy/$(PROJECT).swf \
35             -swf $(library) \
36             -main config/environment.as \
37             -trace logging_helper.log \
38             $(sources)
39
40 .PHONY: console
41 console: open -a SAFlashPlayer scripts/console.swf
```

```
42
43 .PHONY: run
44 run: open -a Safari.app deploy/index.html
45
46 .PHONY: clean
47 clean: rm deploy/*.swf
48
49 .PHONY: production
50 production: $(sources) #$(library)
51 $(MTASC) $(CLASSPATH) \
52 -version $(VERSION) \
53 -out deploy/$(PROJECT).swf \
54 -swf $(library) \
55 -main config/environment.as \
56 -trace no \
57 -exclude config/production_exclude \
58 $(sources)
```

In dem Makefile werden einige Targets (Ziele) definiert, welche die Anwendung in unterschiedlichen Produktionsstatus (development und production) kompilieren und weitere Hilfsaktionen (console, run und clean) ausführen.

## 3.9 Quizaufbau

Nachdem alle verwendeten Programme und Sichten erklärt wurden, kann nun die Struktur der eigentlichen Anwendung vertieft werden.

Die komplette Anwendung befindet sich in dem Ordner „app“, wogegen im Ordner „views“ alle der Anwendung zugehörigen Präsentationselemente definiert werden.

### 3.9.1 Die Präsentationsschicht (View)

Die Präsentationsschicht, die View, befindet sich zum größten Teil in der Flash Quelldatei Library fla. Im Bild 3.5 ist die Struktur der Movieclips erkennbar. Zu Beginn der Studienarbeit gab es keinen Anspruch auf ein variables Design. Aus diesem Grund wurden Buttons, Hintergrundbild und andere gestalterische Elemente wie das Auswahlkästchen direkt in Flash erstellt. Nachdem dann doch ein variables Design gefordert worden war, wurden die Elemente aus der Flashdatei entnommen und mit in den Controller übernommen. Durch die Verschiebung der Designelemente aus der Flashdatei in das Framework, wird das MVC Schema teils durch das Actionpack Schema<sup>3</sup> verändert, so dass die View und der Controller vereint werden. Darauf wird weiter im Abschnitt 3.9.2 eingegangen.

---

<sup>3</sup><http://www.rubyonrails.org/> und [http://de.wikipedia.org/wiki/Ruby on Rails](http://de.wikipedia.org/wiki/Ruby_on_Rails) und [Hei]

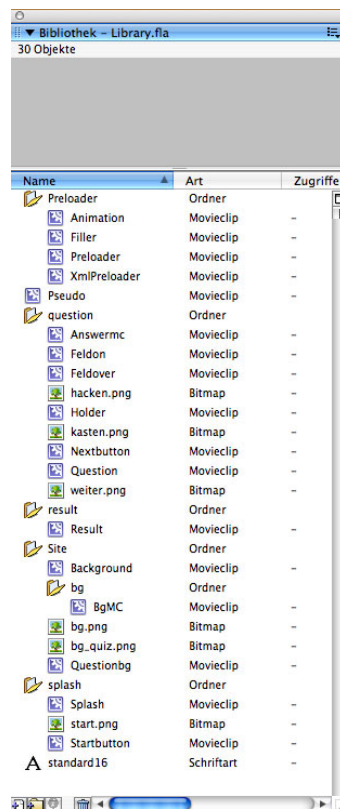


Abbildung 3.5: Übersicht Bibliothek

Das Quiz unterteilt sich in 5 Container (Movieclips):

- Der Preloader (Ebene Preloader), welcher die gesamte Applikation und die XML Datei vorlädt
- Der Hintergrund (Ebene Background), welcher das allgemeine Aussehen definiert
- Die Startseite (Ebene Splash), welche eine Einführung für das Quiz darstellt
- Der Fragenbereich (Ebene Question), welcher die Fragen für die jeweilige Quizart darstellt
- Der Ergebnisbereich (Ebene Result), welcher für die Quizergebnisse zuständig ist

Diese leeren Movieclips geben einen Einstiegspunkt für den jeweiligen Controller. Damit man von außen auf die Movieclips zugreifen kann, müssen diese zugänglich gemacht werden (vgl. ??). Zusätzlich müssen ein eindeutiger Name und Bezeichner definiert werden, welche genauso heißen wie die Controller selbst, und sowohl der Export für Actionscript als auch Export in erstes Bild müsste aktiviert werden. Nachdem alle Movieclips mit diesen Eigenschaften versehen werden, muss der vorläufiger Flashfilm



zur Weiterverwendung veröffentlicht werden. Dieser Flashfilm (Library.swf) befindet sich in dem Ordner „app/views/swf“. In Makefile Zeile 26 (vgl. 3.5) wird dieser Flashfilm definiert und in Zeile 40 bzw. 59 beim kompilieren hinzugefügt.

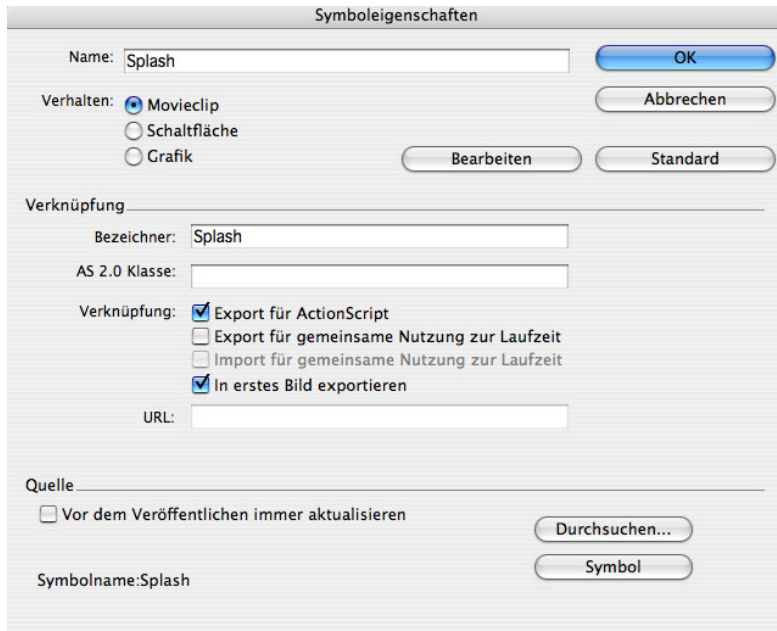


Abbildung 3.6: Freigabe für Actionscript

### 3.9.2 Die Programmschicht (Controller)

Jeder der definierten Views erhält seinen eigenen Controller. Da auf den Preloader ständig zugegriffen wird und den Flashfilm selbst vorladen soll, wird dieser in der Klasse „lib/core/ApplicationControllerBase.as“ in der Zeile 34 initialisiert und ist somit von jedem anderen Controller erreichbar. Die Klasse „Application.as“ ist der Hauptcontroller, der alle anderen Controller verwaltet. Alle Controller zu beschreiben, würde den Umfang der Studienarbeit sprengen. Aus diesem Grund wird das Augenmerk auf den Hauptcontroller (Application) und den Fragencontroller (QuestionController) gelegt.

Im Hauptcontroller werden als allererstes die benötigten Movieclips (vgl. Zeile 6-11 Datei Application.as) und die Controller (vgl. Zeile 15-18 Datei Application.as) definiert. In Zeile 19 wird das Model für die Anwendung erzeugt. Mit der Funktion Application wird der Preloader für den Root, also die Anwendung selbst gestartet, welcher in der Klasse „ApplicationControllerBase.as“ in Zeile 34 definiert wurde. Wie in dieser Klasse beschrieben, wird, nachdem alle Einstellungen und der Preloader geladen wurden, die Funktion didAwake gestartet. Dies geschieht in der Form eines „Callback“ Aufrufes (this, didAwake). Dieser Aufruf stellt sicher, dass alle nötigen Parameter geladen sind, bevor weitere Funktionen aufgerufen werden. Der Aufruf erfolgt mit Hilfe der

importierten Klasse Delegate. Die statische Funktion „create“ in der Klasse „Delegete“ übergibt der Anwendung den Bereich (scope), in dem man sich gerade befindet.

Im weiteren Abschnitt werden die Positionen und die Eigenschaften der Movieclips definiert. Dadurch, dass alle Movieclips initialisiert wurden und anschließend angezeigt werden, werden diese durch die Eigenschaft `_alpha` und `_visible` unsichtbar gemacht und das Model initialisiert. Im Model wird die XML Dateien ausgelesen und alle verwendeten Funktionen definiert. Die genaue Funktion wird im Abschnitt 3.9.3 dargestellt.

Nachdem im Model Funktionen definiert wurden, wird die Funktion `showBackgroundView` aufgerufen. Diese Funktion erzeugt den `BackgroundController` und führt die im Controller definierten Funktionen aus. In diesem Fall werden die definierten Farben vom Model übergeben und der Hintergrund gezeichnet. zum Abschluß der Klasse wird die Funktion `showSplashView` in der Klasse `Application.as` aufgerufen, welche wiederum den `SplashController` erzeugt und die Startseite anzeigt.

In dem `SplashController` sind 3 Funktionen definiert. `didAwake` ist immer die erste Funktion, welche in einem Controller aufgerufen wird. Als alle erstes werden die definierten Farben importiert, die Textfelder für Fragen und Beschreibung dynamisch erstellt und mit Inhalt gefüllt. Alle Inhalte werden über die im Model definierten Funktionen übergeben. Die Funktion `onReleaseStart` reagiert auf den Startbutton. Wenn dieser betätigt wird, dann wird eine weitere Funktion ausgeführt, welche die Funktion `showQuestionView` im Hauptcontroller (`Application.as`) startet und alle Elemente des `SplashController` löscht.

Nachdem die Funktion aufgerufen wurde, startet das eigentliche Quiz. In der Funktion `didAwake` des `QuestionController` werden als erstes die Farben definiert und die verwendeten Buttons erstellt. Die Funktion `onQuizStart`, welche bei jeder Frage und Art des Quizzes aufgerufen wird, definiert die allgemeine Struktur; hier wird die Frage, der Punktestand und das aktuelle Feedback aufgezeigt. Gleichzeitig wird durch die Funktion `isForm` die in der XML Datei definierte Art des Quizzes ermittelt und die entsprechende Funktion in einer Schleife der Anzahlentsprechend aufgerufen.

- `createMcAnswerArray` - Erstellung des Multiplechoice Tests
- `createDndAnswerArray` und `createDragArea` - Erstellung des Drag And Drop Tests
- `createImageAnswerArray` - Erstellung des Multiplechoice Tests mit Bildern
- `createDndImageAnswerArray` und `createImageDragArea` - Erstellung des Drag and Drop Tests mit Bildern

Zur Verdeutlichung des Aufbaus wird die Art „Multiple Choice Quiz“ und somit die Funktion `createMcAnswerArray`, als auch „Drag And Drop Quiz“ also die Funktion `dragAnswerOnRelease` in Augenschein genommen.

Die Funktion `createMcAnswerArray` definiert die ausgelesenen Fragen. Hierbei werden alle Elemente wie Auswahlkästchen und Textfelder so oft erstellt, wie es Antworten gibt. Für jede Antwort wird die Funktion `answerOnRollover`, `answerOnRollout`

und `answerOnRelease` zugewiesen, welche für die Aktionen Rollover (Mauszeiger wird über den Text bewegt), Rollout (Mauszeiger wird vom Text wegbewegt) und Release (Drücken auf den Text) zuständig sind. Bei der letzten Aktion wird geprüft, ob die richtige bzw. falsche Antwort gewählt wurde. Das Feedback wird sofort angezeigt und eine weitere Korrektur der Antwort ist nicht möglich.

Das Drag and Drop Quiz ist äquivalent zu dem Multiple Choice Quiz. Bei dieser Art werden die Fragen nicht durch Anklicken der Antwort sondern durch „ziehen und loslassen“ der Antwort in einem vordefiniertem Bereich beantwortet. Die Quizarten, welche auf Bildern basieren funktionieren entsprechend.

Nachdem alle Fragen beantwortet wurden, wird der `ResultController` mittels der Funktion `showResultView` aufgerufen. Durch diese Funktion wird die View initialisiert und stellt die Ergebnisseite dar.

Um die Interaktion zu verbessern und das Aussehen zu verschönern wurden einige Open Source Klassen eingesetzt. Die externen Klassen, welche sich im Ordner „lib/apis“ befinden, werden durch den Befehl `import` oberhalb der Klasse importiert. In der Anwendung wurde vor allem die Klasse `Delegate`<sup>4</sup> für die Übergabe des aktuellen Arbeitsbereichs, Klasse `MovieClipUtil` für Übergabe der richtigen Reihenfolge der Movieclips und die Klasse `Tooltip`, welche bei den Bildquizzes ihren Einsatz findet, verwendet. Die Klasse `tweenManager`<sup>5</sup> wird in der kompletten Anwendung verwendet und direkt ins Framework in der Klasse „`ApplicationControllerBase.as`“ definiert und in der Konfigurationsdatei „`environment.as`“ geladen.

### 3.9.3 Das Datenmodell (Model)

Das Model enthält die darzustellenden Daten. Alle Daten werden aus der XML Datei geladen und für die Kontroller aufbereitet. Durch die `set` Funktionen werden die Daten aus der XML Datei definiert und durch die `get` Funktionen werden diese Daten von den Kontroller angefordert.

### 3.9.4 Softwaretechnische Sicht

Anhand eines Klassendiagramms (vgl. Bild 3.7) kann die Struktur der Anwendung veranschaulicht werden.

---

<sup>4</sup><http://dynamicflash.com/2005/05/delegate-version-101/>

<sup>5</sup><http://laco.wz.cz/tween/>

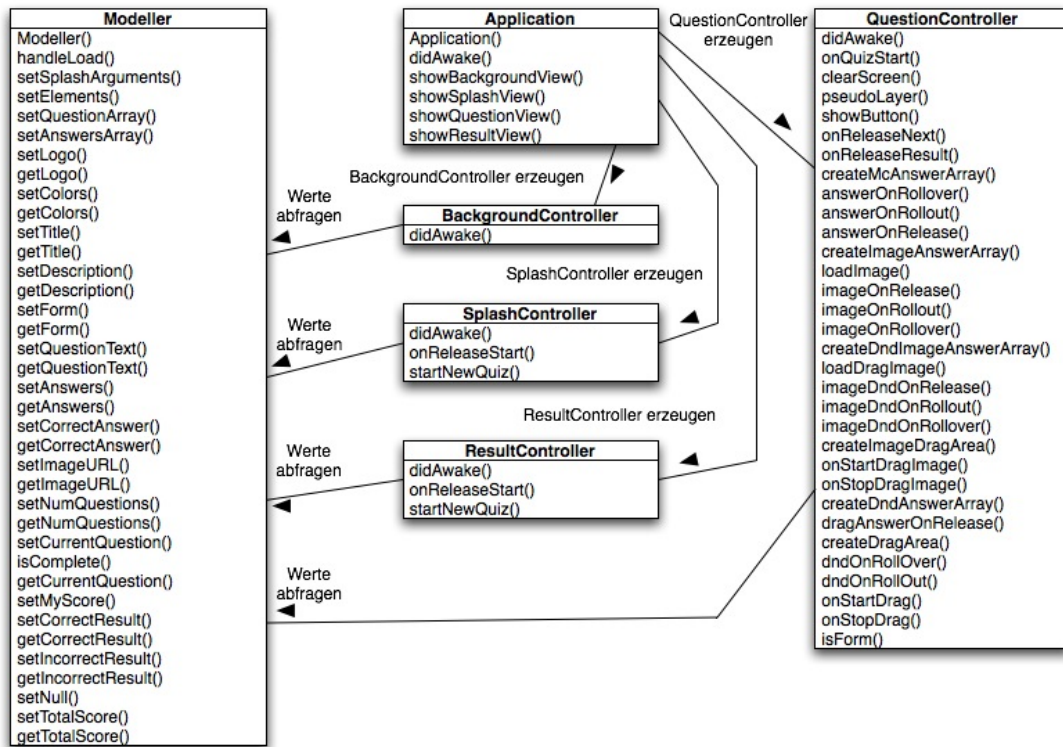


Abbildung 3.7: Klassendiagramm

## 4 Ausblick

Bei der Erstellung der Anwendung wurde sehr auf die Modularität geachtet. Diese Eigenschaft erleichtert es, das Programm zu erweitern und zu verbessern. Dadurch wäre es möglich, Quizarten wie „Fill in the Blank“, „Hot Objects“, „Hot Spot“ oder „True or False“ einzubinden. Durch den Einsatz des Open Source Kompilers MTASC und weiteren Open Source Klassen ist es für jeden Programmierer möglich, die Anwendung weiter zu entwickeln. Das einzige Hindernis ist der Einsatz der kommerziellen Software Adobe Flash. Dieses Problem könnte jedoch mit der Verwendung des Open Source XML Prozessors swfmill<sup>6</sup> behoben werden. Mit swfmill besteht die Möglichkeit, in XML Struktur geschriebenen Code (vgl. 4.1) in SWF-Film generieren zu lassen. Der kommandozeilenbasierte Verarbeiter erhält im Gegensatz zu Flash keine IDE, stellt jedoch die gleichen Eigenschaften zur Verfügung.

Listing 4.1: Beispiel swfmill XML Datei

```
1 <?xml version="1.0" encoding="uft-8"?>
2 <movie width="320" height="240" framerate="30">
3   <background color="#ffffff"/>
4     <frame>
5       <library>
6         <clip id="spheres" import="library/spheres.png"/>
7       </library>
8     </frame>
9 </movie>
```

Eine weitere Erweiterung des Quizzes wäre die Integration in eine bestehende eLearning Plattform. Durch diese Modifikation wäre es einfacher und schneller ein neues Quiz zu erstellen und die vorhandenen Quizes zu verwalten. Ein Schritt in die Richtung wäre, als erstes die XML Datei für eine eLearning Plattform zu standardisieren (z.B. IMS QTI).

### 4.1 Standards

Durch den zunehmenden Einsatz der eLearning Angebote und einen immer größer werdenden Anbietermarkt ist es wichtig, die Daten zu standardisieren, um einen einfachen Umstieg oder die Verwendung unterschiedlicher Plattformen zu ermöglichen. Hierbei

---

<sup>6</sup><http://www.swfmill.org>

haben sich mehrere Standardisierungskonsortien gebildet, welche die Standards definieren.

Das SCORM (Sharable Content Object Reference Model) der Organisation ADL (Advanced Distributed Learning) gehört hierbei zu den wichtigsten Standards im eLearning Umfeld. Diese Organisation versucht die Arbeit mehrerer Standardisierungsgruppen wie die AICC (Aviation Industry Computer-Based Training Committee), DCMI (Dublin Core Meta-data Initiative), IEEE (Institute of Electrical and Electronics Engineers), IMS (Instructional Management System) und weitere zusammenzuführen [Wikb]. Durch diese Zusammenführung soll ein universelles Inhalt-Modell erstellt werden.

#### **4.1.1 IMS QTI**

Eine speziell für Tests entwickelte standardisierte Datenformat das IMS QTI wurde von IMS (Instructional Management System) entwickelt. Mit Hilfe dieser Standards ist der Austausch von Tests und Quizzes zwischen unterschiedlichen Anwendungen möglich. Wenn das Quiz an diesen Standard angepasst werden würde, dann könnte man die Anwendung als ein Plugin für mehrere unterschiedliche eLearning Plattformen ansehen.

## 5 Zusammenfassung

Zu Beginn der Studienarbeit sollte ein Quiz mit Hilfe von Adobe Flash eLearning Komponenten erstellt werden. Durch die feste Integration der eLearning Komponenten in die Flash IDE, wäre es sehr umständlich gewesen, diese Komponenten für den Einsatz mit XML Dateien umzuschreiben. Aus diesem Grund wurde ein Framework und der Open Source Flash Compiler MTASC verwendet, um die Anwendung zu erstellen. Somit wurden zwei unterschiedliche Quizzarten „Multiple Choice “ und „Drag and Drop “ erstellt. Zusätzlich wurde nicht nur eine Textvariante entwickelt, sondern zusätzlich die beiden Quizzarten in Form eines Bildquizzes implementiert. Diese Anwendung kann jetzt für das Projekt Feuerwehr- und Katastrophenschutzschule Rheinland - Pfalz (LFKS) und für andere Projekte eingesetzt werden. Die Anwender haben die Möglichkeit, Fakten und Anwendungswissen bezüglich der Ausbildung bei der Feuerwehr zu vertiefen und ihr Wissen zu erproben.

## Literaturverzeichnis

- [Aka] Akademiehandel: *Vorteile durch E-Learning*.  
URL <http://www.akademie-handel.de/elearning1.php?item=5>
- [Dah] Dr. I. Dahn: *eLearning-Ausbau in Koblenz - das Campus-Projekt MOSEL*, 26. April 2005.  
URL <http://iwm.uni-koblenz.de/elt3dvd/tag1.html#vortrag04>
- [Hei] David Heinemeier: *Rails*, 25. Juli 2004.  
URL <http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-talk/107370>
- [IfW] Universität koblenz Institut für Wissensmedien: *Projektbeschreibung Feuerwehr*, 11. November 2005.  
URL <http://iwm.uni-koblenz.de/iwm/forschung/projekte/feuerwehr>
- [Wika] Wikipedia: *eLearning*, 25. Juli 2006.  
URL <http://de.wikipedia.org/wiki/E-Learning>
- [Wikb] Wikipedia: *Scorm*, 12. Juni 2006.  
URL <http://de.wikipedia.org/wiki/Scorm>