HOCHSCHULE COBURG

Hochschule für angewandte Wissenschaften Coburg

Fakultät Elektrotechnik und Informatik

Studiengang:

Informatik (Bachelor)

Evaluation der Verwendung von Scalable Vector Graphics in der iELP

Hausarbeit

Lukas Höllein

Abgabe der Arbeit: 09. Januar 2017

Betreut durch:

Prof. Dr. Dieter Wißmann, Hochschule Coburg

im Rahmen des Studienfachs Wissenschaftliches und interdisziplinäres Arbeiten

Inhaltsverzeichnis

Abbildungsverzeichnis

Codebeispielverzeichnis

Abkürzungsverzeichnis

iELP	Interaktive E-Learning Plattform
SVG	Scalable Vector Graphics (als Abkürzung im Singular benutzt)
HTML	Hypertext Markup Language
XML	Extensible Markup Language
W3C	World Wide Web Consortium
CSS	Cascading Style Sheets
SMIL	Synchronized Multimedia Integration Language
D3	Data Driven Documents

1 Einleitung

Dieses Dokument stellt die Hausarbeit im Studienfach *Wissenschaftliches und interdisziplinäres Arbeiten* im Wintersemester 16/17 dar und bildet neben der gleichnamigen Abschlusspräsentation die Abgabe für dieses Fach.

Im Folgenden wird die Hausarbeit näher vorgestellt.

1.1 Kontext

Der Kontext für diese Hausarbeit wurde vom Betreuer vorgegeben. Ein Entwicklerteam soll im Rahmen eines Projekts eine bestehende Webanwendung, die sogenannte interaktive E-Learning Plattform (iELP) weiterentwickeln. Diese bietet die Möglichkeit, Wissen interaktiv auf einer Website bereitzustellen und zu trainieren. In nachstehender Abbildung ist die Startseite der iELP zu sehen.

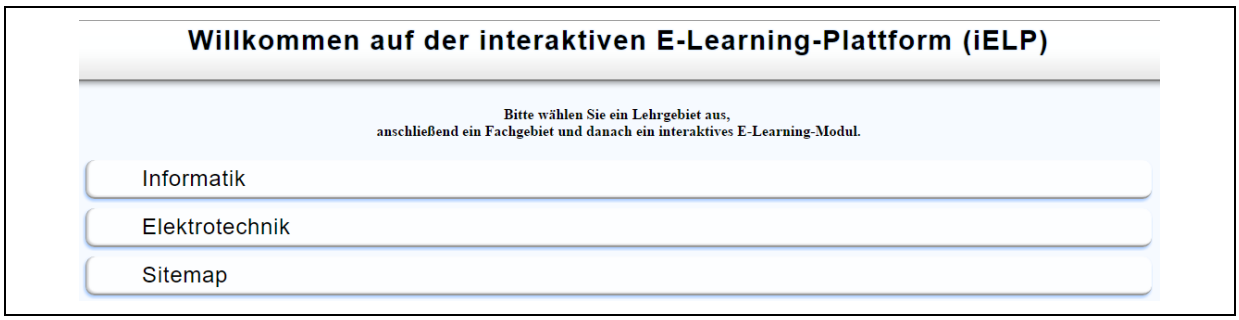

Abbildung 1: Ausschnitt der Startseite der iELP
aus (Hochschule Coburg, 2016)

Das Entwicklerteam überlegt, ob zusätzliche Techniken und Tools sinnvoll für die iELP verwendet werden können. Jedem Mitglied des Entwicklerteams wurde dazu eine Möglichkeit zur Verbesserung zugeteilt, welche untersucht werden soll. Die Hausarbeit stellt dabei die Dokumentation dieser Untersuchung dar.

In dieser Arbeit wird die Verwendung von Scalable Vector Graphics (SVG) auf der iELP untersucht und beurteilt.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist es, eine Einschätzung zu geben, ob mit SVG die iELP verbessert werden kann, um auf dessen Basis nachfolgend zu entscheiden, ob ein tatsächlicher Einsatz sinnvoll ist und durchgeführt wird.

Dafür ist zunächst zu definieren, welche expliziten Verbesserungen erreicht werden sollen. Diese werden im nachfolgenden Kapitel erläutert.

1.3 Fragestellungen

Diese Arbeit wird insgesamt vier Fragestellungen untersuchen und beantworten. Diese sollen alle Facetten, die von SVG geboten werden und für die iELP einen Mehrwert bieten könnten, beurteilen, um die genannte Zielsetzung zu erfüllen.

Die vier Fragen sind:

1. Kann SVG die Nutzerfreundlichkeit durch seine Features steigern?
2. Kann SVG Veranschaulichungen darstellen und in welchem Maße vorhandene ersetzen?
3. Kann SVG animierten Inhalt sinnvoll gestalten?
4. Kann SVG durch passende Entwicklungswerkzeuge einfach entworfen werden?

Auf die Fragen wird im Laufe der Arbeit Bezug genommen und diese werden beantwortet.

1.4 Aufbau der Arbeit

Die Arbeit gliedert sich zunächst in das Kapitel *Grundlagen der Scalable Vector Graphics*, welches den Lesern zunächst einen Überblick verschafft, worum es sich bei SVG handelt, wie es funktioniert und was es für Features anbietet. Dies ermöglicht es, die Aussagen und Einschätzungen zu SVG detailliert nachzuvollziehen und bildet den größten Teil der Arbeit.

Anschließend greift das Kapitel *Einsatzmöglichkeiten in der iELP* diese Erläuterungen auf und bringt konkrete Beispiele, wie SVG auf der iELP eingesetzt werden kann. Dies ist zugleich auch der erste Schritt, um die Fragestellungen zu beantworten.

Zum Abschluss wird das Kapitel *Fazit* die Fragestellungen endgültig beantworten und die in der Zielsetzung geforderte Einschätzung geben. Das Kapitel *Ausblick* schildert grob die nächsten Schritte, die im Rahmen des Kontexts der Arbeit nötig sein werden.

Einige Fachbegriffe, die im normalen Textfluss nicht erklärt werden, sind zudem im *Glossar* kurz erläutert und Abkürzungen befinden sich ausformuliert im *Abkürzungsverzeichnis*.

Im *Anhang* sind alle vorgestellten Codebeispiele als lauffähiger Quellcode vorzufinden, die zum Testen verwendet werden können.

2 Grundlagen der Scalable Vector Graphics

Scalable Vector Graphics (SVG) bietet die Möglichkeit grafische Informationen und Bilder in einer kompakten und portablen Form zu repräsentieren, also diese intern zu notieren und aufbauend auf diesen Daten eine Darstellung zu erzeugen. Um diese Informationen zu speichern, benutzt SVG das XML-Format (Eisenberg & Bellamy-Royds, 2014, S. 1), wodurch es lesbar und einfach zu verstehen ist (Dailey, Frost, & Strazzullo, 2012, S. 3). Es ist zudem ein Standard für grafische Darstellungen im Internet, spezifiziert durch das World Wide Web Consortium (W3C). Somit bietet es nicht nur eine einheitliche Form, um geometrische Figuren zu notieren, sondern ist zudem ein populäres und leicht einsetzbares Werkzeug für Anwendungen im World Wide Web, die weit über die rein statische Darstellung von Grafiken hinaus gehen können (Dailey, Frost, & Strazzullo, 2012, S. 1).

Der entscheidende Vorteil von SVG gegenüber anderen Möglichkeiten Grafik zu repräsentieren, ist dessen unendliche Skalierbarkeit. Auf dieses Thema wird im nachfolgenden Kapitel näher eingegangen.

Jegliche grafische Information, wie zum Beispiel ein Kreis oder ein Rechteck, wird bei SVG in einem zweidimensionalen Koordinatensystem gespeichert (Eisenberg & Bellamy-Royds, 2014, S. 1), das wie in Abbildung 2 zu sehen, aufgebaut ist.

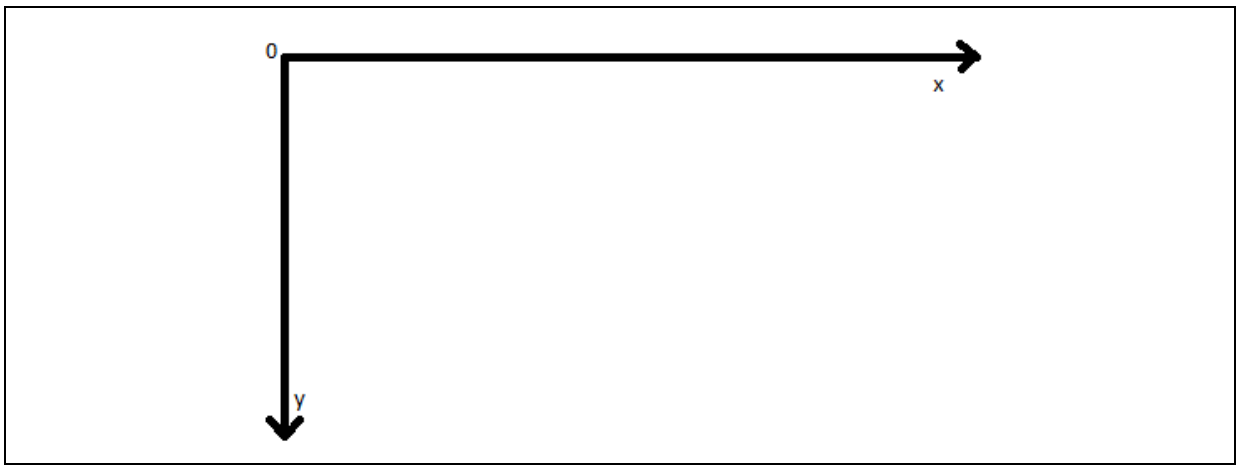

Abbildung 2: Aufbau des Koordinatensystems von SVG

Das bereits angesprochene XML-Format zur Notation von SVG ist ebenfalls stark an HTML angelehnt, wodurch SVG Dateien, die mit der Endung `.svg` versehen werden, mithilfe des HTML-Tag Konzepts aufgebaut sind (W3C SVG Working Group, 2011). Beispiele solcher SVG Dateien finden sich in verschiedenen Teilen der Arbeit, sowie im Anhang und können dort eingesehen werden.

Da das Speichern und Darstellen von grafischen Informationen heutzutage auf sehr viele Arten möglich ist und die Verwendung eines Koordinatensystems wie hier geschildert nur ein bestimmtes Konzept ist, wird nun zunächst die grundlegende Funktionsweise der Darstellung von Grafik bei SVG erläutert, gegenübergestellt und beurteilt.

2.1 Vektorgrafiken und Rastergrafiken

Wie der Name Scalable Vector Graphics schon vermuten lässt, basiert die Darstellung von Grafik in SVG auf sogenannten Vektorgrafiken. Neben Vektorgrafiken sind Rastergrafiken ebenfalls ein weit verbreitetes Mittel, um Bilder oder grafische Informationen zu repräsentieren. Zusammen bilden sie die zwei wichtigsten Möglichkeiten, um auf Computern solche Darstellungen durchzuführen (Eisenberg & Bellamy-Royds, 2014, S. 1).

Bei Rastergrafiken wird ein Bild als zweidimensionales Array von Pixeln repräsentiert (s. Abb. 3), welchen eine bestimmte Farbe zugeordnet ist (Eisenberg & Bellamy-Royds, 2014, S. 1).

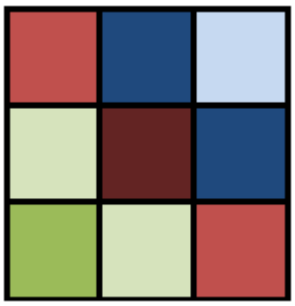

Abbildung 3: Eine 3x3 Rastergrafik

Bei Vektorgrafiken wird ein Bild als eine Ansammlung von geometrischen Figuren in einem Koordinatensystem beschrieben, wodurch es kein fertiges Array aus Pixeln ist, sondern als Anweisung zur Darstellung von Figuren an bestimmten Positionen verstanden werden kann (Eisenberg & Bellamy-Royds, 2014, S. 2). In Bezug auf das Koordinatensystem aus SVG, lässt sich das Beispiel aus Abbildung 3 als Vektorgrafik etwa wie folgt interpretieren:

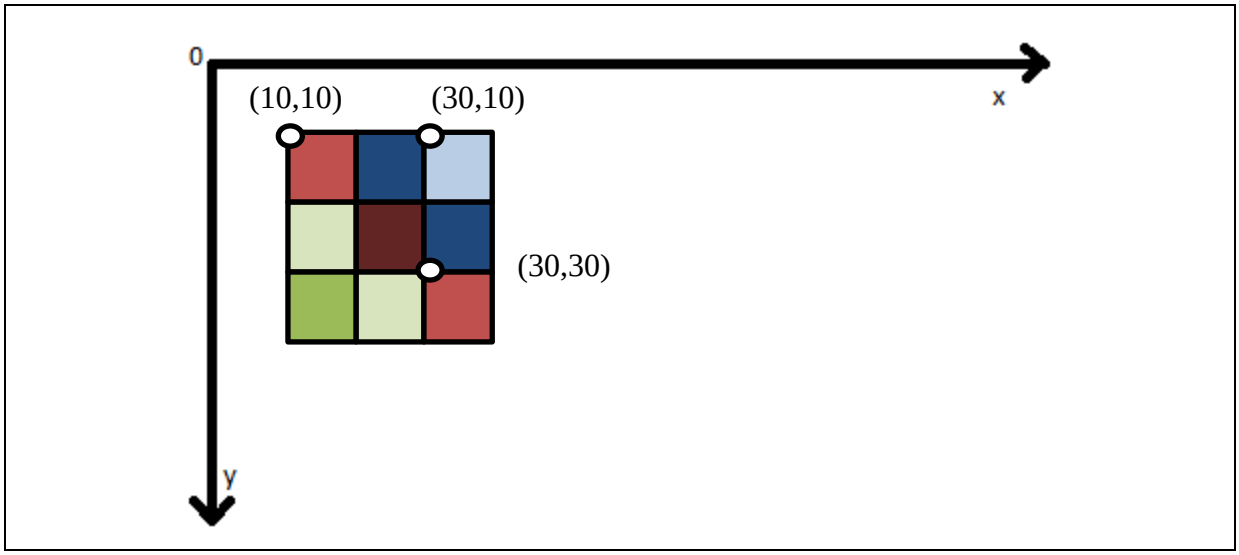

Abbildung 4: Interpretation einer Vektorgrafik mit Koordinatenpunkten

In (Eisenberg & Bellamy-Royds, 2014, S. 2) wird der wesentliche Unterschied der zwei Technologien gut auf den Punkt gebracht: *„[…] raster graphics work by describing which squares should be filled in with which colors […] Vector graphics 'understand' what they are – a square 'knows' it's a square […]".*

Waren die bunten Rechtecke als Rastergrafik in Abb. 3 nur gefärbte Bereiche, so sind diese als Vektorgrafik in Abb. 4 nun wirklich als Rechtecke im mathematischen Sinne definiert, denn diesen wurde eine Position in Koordinaten, sowie eine Breite und Höhe zugewiesen.

Anhand dieses Unterschieds lässt sich nun der entscheidende Vorteil von Vektorgrafiken gegenüber Rastergrafiken ableiten: Skalierbarkeit (Eisenberg & Bellamy-Royds, 2014, S. 3).

Ein häufiger Anwendungsfall im Internet oder generell bei der Betrachtung von Bildern ist es diese zur besseren Erkennung zu vergrößern. Der daraus resultierende Sachverhalt ist an (Eisenberg & Bellamy-Royds, 2014, S. 4f) angelehnt. Beispielsweise bleibt einer Rastergrafik bei einer Vergrößerung um den Faktor 4 nichts Anderes übrig, als jeden einzelnen Pixel 4-mal so groß darzustellen. Dies hat Nachteile wie das „Verpixeln" und Unschärfe von Bildern, also einen Qualitätsverlust, zur Folge, was durch diverse Techniken wie sogenanntem Antialiasing zwar verbessert, jedoch nicht behoben werden kann.

Vergrößert man jedoch eine Vektorgrafik ebenfalls um den Faktor 4, muss das darstellende Medium, wie etwa ein Bildschirm, nur alle einzelnen mathematischen Figuren, ebenfalls mit dem Faktor 4 multipliziert, darstellen. Da es sich dabei um vektorbasierte Figuren handelt, werden dafür lediglich die Koordinaten dieser verändert, also skaliert. Dies führt zu keinerlei

„Verpixeln“, sondern zu nach wie vor gleich scharfen Darstellungen, da die Figuren nun von Grund auf größer definiert und dargestellt werden.

Um auf Webseiten im Internet ein Bild vergrößert darzustellen, müssen zudem keine neuen Daten an den Client übermittelt werden, sondern es genügt, wenn der Client die Vergrößerung wie oben erwähnt selbständig auf Basis des ursprünglichen Bilds durchführt (Dailey, Frost, & Strazzullo, 2012, S. 3).

Abbildung 5 soll diesen fundamentalen Unterschied veranschaulichen.

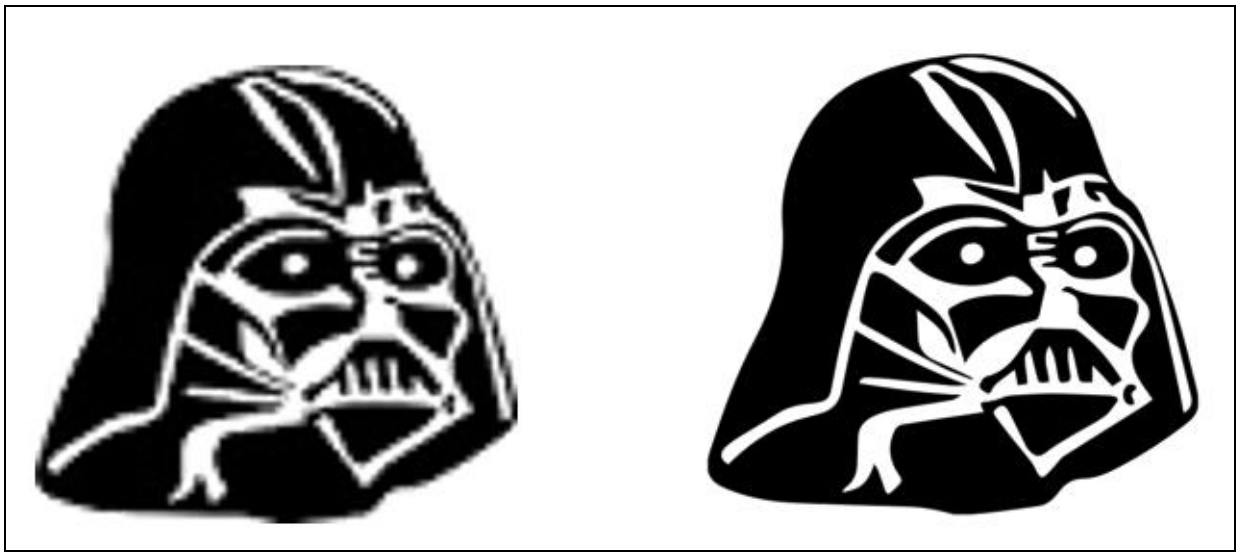

Abbildung 5: Nach Vergrößerung: Links eine Rastergrafik (verpixelt), rechts eine Vektorgrafik (scharf)
aus (TheCraftChop, 2016)

SVG repräsentiert alle Grafiken als Vektorgrafiken und kann somit von dem Vorteil der unendlichen Skalierbarkeit profitieren.

Nachdem nun die generelle Funktionsweise von SVG erklärt ist, wird anschließend dargestellt, wie genau einzelne geometrische Figuren erzeugt werden können, die dann eine zusammenhängende Vektorgrafik bilden.

2.2 Features

Die Hauptaufgabe von SVG ist, wie bereits erwähnt, das Repräsentieren von grafischen Informationen. Einen guten Überblick über alle Features von SVG bietet (W3C SVG Working Group, 2011).

Um in SVG Grafik zu erzeugen, gibt es die Möglichkeit verschiedene geometrische Figuren (im Fortlaufenden auch nur *Figuren* genannt) zu notieren und diese mit verschiedenen Eigenschaften wie zum Beispiel Größe und Farbe zu versehen.

Neben geometrischen Figuren bietet SVG auch Filter, wie zum Beispiel einen Schwarz-Weiß Filter an, der die einzelnen Bestandteile entsprechend anders darstellt. Des Weiteren können neben statischen, unbeweglichen Figuren diese auch auf verschiedene Art animiert und bewegt werden. Außerdem besitzt SVG ein eigenes Document Object Model (DOM), welches an das bestehende HTML DOM anknüpft, wodurch SVG über CSS und JavaScript manipuliert werden kann.

Die einzelnen Features, die hier nur kurz aufgezählt wurden, werden in den kommenden Kapiteln nun näher beschrieben.

2.2.1 Geometrische Figuren

Die geometrischen Figuren, die zur Verfügung stehen, sind Rechtecke, Kreise, Ellipsen, Linien, Polylinien, Polygone und Pfade. Texte können mithilfe von SVG ebenfalls dargestellt werden. Jede dieser Figuren wird über einen eigenen Tag erzeugt, in dem alle Informationen zur jeweiligen Figur spezifiziert werden. Genaue Details zu allen Attributen der einzelnen Figuren, sowie Erklärungen finden sich übersichtlich in (W3C SVG Working Group, 2011), (Eisenberg & Bellamy-Royds, 2014, S. 39-54) und (Dailey, Frost, & Strazzullo, 2012, S. 32-51) gegliedert. Im Folgenden werden einzelne Figuren und deren Syntax zur Verdeutlichung daraus gezeigt.

Beispielsweise wird ein Kreis mit dem Tag `<circle>` beziehungsweise `</circle>` erzeugt und geschlossen. Im Codebeispiel 1 ist eine vollständige Definition für einen Kreis zu sehen, der mit einigen Attributen erstellt wurde.

```
<circle cx="50" cy="50" r="40" stroke="black" stroke-
width="3" fill="red" />
```

Code 1: Eine Kreisdefinition mit einigen Attributen

Die Attribute `cx` und `cy` geben den Mittelpunkt des Kreises als Koordinatenpunkt im Format `(cx, cy)` an, welcher im Koordinatensystem von SVG platziert wird. Das Attribut `r` bezeichnet den Radius des Kreises, `stroke` gibt die Farbe des Kreisrahmens an, `stroke-width` dessen Dicke und `fill` gibt die Füllfarbe an.

Die Einheit der numerischen Werte entspricht in diesem Fall der HTML Einheit `px`, aber es können auch die Einheiten %, `em` und `rem` gewählt werden. Generell ist SVG sehr kompatibel

mit HTML Attributen und Elementen. Auf genaue Details dieser Verknüpfung wird im Kapitel *Einbettung und Umsetzung* genauer eingegangen und soll hier nur kurz erwähnt bleiben.

Ein Kreis, wie oben definiert, ergibt die in Abbildung 6 zu sehende Darstellung, wenn man sie zur Anzeige an einem Bildschirm bringt.

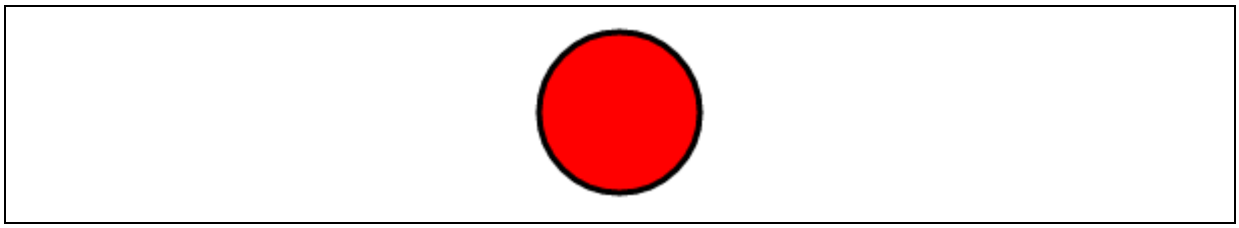

Abbildung 6: Darstellung der Kreisdefinition aus Code 1

Eine weitere geometrische Figur in SVG ist der Pfad, der mit dem Tag `<path>` beziehungsweise `</path>` erzeugt und geschlossen wird. Damit können Linien und Kurven beliebiger Form und Länge erstellt werden, was die Darstellung von fast allen erdenklichen Formen ermöglicht. Nachfolgend ist die Definition eines einfachen Pfads zu sehen.

```
<path d="M 100 200 q 150 -300 300 0" stroke="blue" stroke-
width="5" fill="none" />
```

Code 2: Eine Pfaddefinition mit einigen Attributen

Die im Attribut `d` vorhandenen Buchstaben und Zahlen geben den Verlauf des Pfades an. Der Buchstabe `M` bewegt den Pfad auf die nachfolgenden Koordinaten `(x, y)` des Koordinatensystems von SVG. Der Buchstabe `q` gibt den Verlauf einer sogenannten quadratischen Bézier Kurve an, die unter (Dailey, Frost, & Strazzullo, 2012, S. 42) näher beschrieben wird. Pfade in SVG besitzen noch einige weitere Möglichkeiten, den Verlauf anzugeben. Der Pfad aus Codebeispiel 2 ist in Abbildung 7 dargestellt.

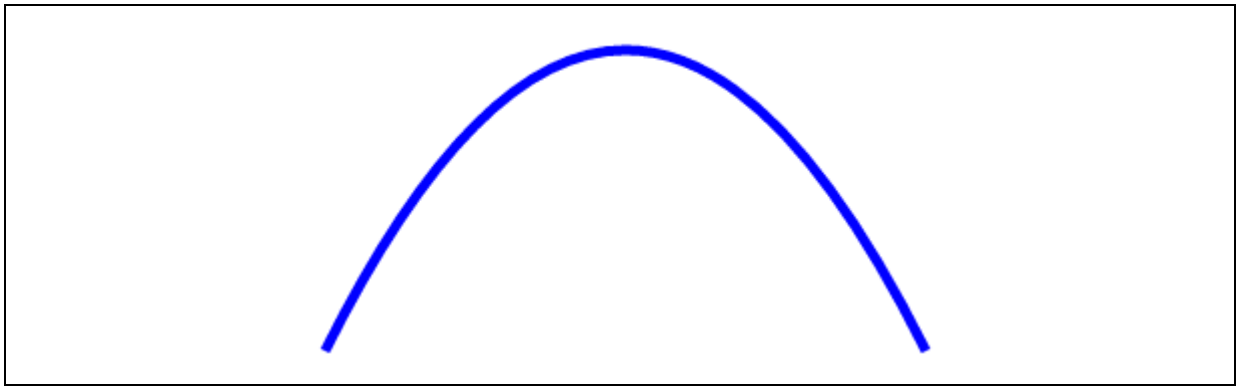

Abbildung 7: Darstellung der Pfaddefinition aus Code 2

Dieses Beispiel soll verdeutlichen, wie auf einfache Art Formen jeglicher Komplexität erstellt werden können. Beispielsweise ist die Vektorgrafik aus Abbildung 5 ebenfalls eine SVG Datei, die mit mehreren Pfaden definiert wurde. Diese kann im Anhang eingesehen werden. Pfade sind somit ein geeignetes Mittel um Grafiken, die über einfache Formen wie Kreise hinausgehen, zu erstellen.

Dieses Kapitel sollte einen detaillierteren Einblick in die Erstellung einzelner Figuren geben, wenngleich es nicht möglich ist, jede der genannten anzusprechen. Aus den Beispielen soll jedoch hervorgehen, dass die Syntax in SVG relativ leicht und verständlich ist und somit schnell erlernt werden kann.

In den nachfolgenden Kapiteln wird erklärt, wie Figuren mit zusätzlichen Mitteln verändert werden können.

2.2.2 Animationen und Filter

SVG bietet die Möglichkeit Figuren mithilfe von Animationen zu bewegen und diese mit Filtern anders darzustellen. Nachfolgend soll mit zwei Beispielen diese Fähigkeit demonstriert werden. Genauere Informationen dazu finden sich wiederum in (W3C SVG Working Group, 2011), (Eisenberg & Bellamy-Royds, 2014, S. 155-207) und (Dailey, Frost, & Strazzullo, 2012, S. 89-100, 146-177).

Animationen werden unter anderem mithilfe des `<animateMotion>` beziehungsweise `</animateMotion>` Tags direkt innerhalb einer Figur definiert und geschlossen. Dies bietet eine deklarative Möglichkeit Animationen zu erstellen, die aus einem anderem W3C Standard entnommen wurde, der Synchronized Multimedia Integration Language (SMIL) (Dailey, Frost, & Strazzullo, 2012, S. 89).

Bewegungen jeglicher Art sind denkbar. Mithilfe des untenstehenden Codebeispiels wird ein Kreis entlang einer Strecke bewegt.

```
<circle cx="50" cy="50" r="50" fill="red">
     <animateMotion dur="5s"
          values="0,0; 500,0; 250,250; 0,0"
          repeatCount="indefinite" />
</circle>
```

Code 3: Ein Kreis, der entlang einer vorgegebenen Strecke bewegt wird

Der `animateMotion` Tag nimmt dabei das Attribut `dur` entgegen, welches angibt, wie lange der Kreis insgesamt in Sekunden für eine vollständige Bewegung brauchen soll. Das Attribut `values` nimmt eine unbegrenzte Kette an Punkten entgegen, entlang derer der Kreis sich von Punkt zu Punkt bewegt. Schließlich gibt das Attribut `repeatCount` an, dass die Animation unendlich oft hintereinander ausgeführt werden soll. In Abbildung 8 wird angedeutet, wie sich dieser Kreis verhalten wird.

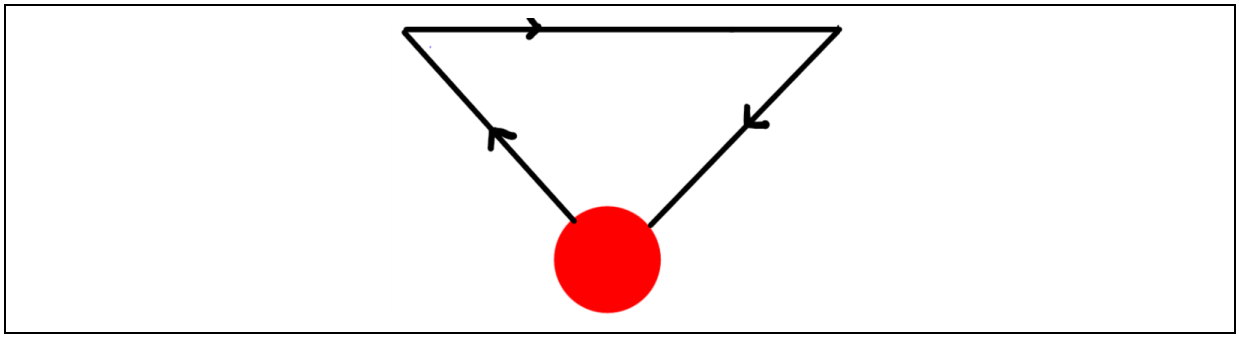

Abbildung 8: Angedeuteter Verlauf des Kreises aus Code 3

Neben `animateMotion` gibt es noch andere Arten von Animationen, wie zum Beispiel `animateTransform`, die hier jedoch nicht erläutert werden.

Filter werden in SVG mit dem `<filter>` beziehungsweise `</filter>` Tag erstellt und geschlossen und werden im Gegensatz zu Animationen außerhalb von Figuren erstellt. Diese verweisen dann auf den gewünschten Filter, um ihn zu verwenden. Es werden eine Vielzahl an Filtern angeboten, nachfolgend wird ein Beispiel vorgestellt, welches den sogenannten *GaussianBlur* Filter benutzt, der Unschärfe erzeugt.

```
<filter id="name">
    <feGaussianBlur stdDeviation="5" />
</filter>
<circle cx="100" cy="100" r="50" fill="green"
       filter="url(#name)" />
```

Code 4: Ein Filter wird auf einen Kreis angewendet

Dem Filter wird zunächst eine `Id` zugewiesen, wie es für alle HTML Elemente auch üblich ist. Im Inneren des Filters wird definiert, wie er sich verhalten soll. Hier wird das Element `feGaussianBlur` verwendet, welches mit dem Attribut `stdDeviation` versehen wird, was den Unschärfegrad bestimmt. Anschließend wird dem Kreis über das `filter` Attribut der Filter anhand seiner `Id`, wie oben zu sehen, zugewiesen.

In Abbildung 9 wird der oben erzeugte Kreis einem Kreis ohne Filter gegenübergestellt, um den Effekt zu verdeutlichen.

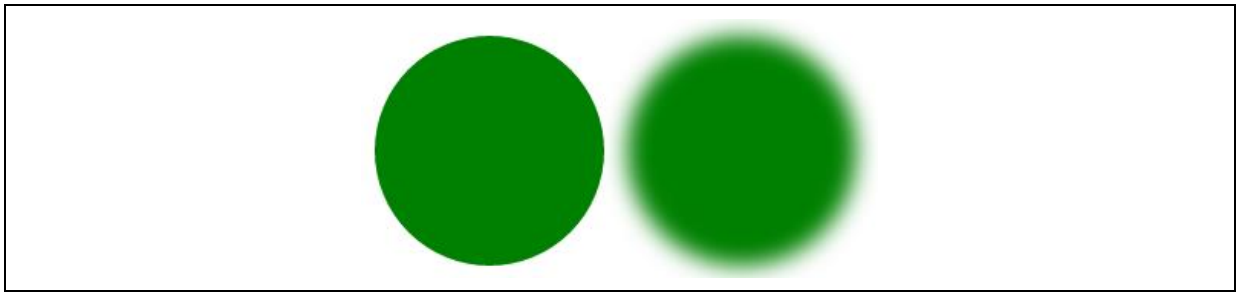

Abbildung 9: Links ein Kreis ohne GaussianBlur, rechts mit GaussianBlur

Dieses Kapitel sollte verdeutlichen, dass SVG neben der bloßen Erzeugung auch Mittel bietet, um mit geometrischen Figuren weitergehend zu arbeiten.

2.2.3 SVG mit JavaScript und CSS

SVG verfügt über ein eigenes DOM, welches das DOM von HTML komplett unterstützt und darauf aufbaut (W3C SVG Working Group, 2011). Dies ermöglicht es, jedes SVG Element mit CSS und JavaScript auf bekannte Art und Weise anzusprechen und zu manipulieren, wie alle anderen HTML Elemente auch (Dailey, Frost, & Strazzullo, 2012, S. 3).

Damit kann jede Vektorgrafik auch interaktiv gestaltet werden, wodurch SVG eng an bestehende HTML Infrastruktur angebunden werden kann. Die zwei nachfolgenden Beispiele sollen diese Fähigkeit demonstrieren.

Zuerst ist eine CSS Anweisung zu sehen, die für alle Kreise die Farbe und Rahmendicke festlegt, sodass diese nicht bei jedem einzelnen Kreis festgelegt werden muss.

```
circle{
     fill: blue;
     stroke-width: 5;
}
```

Code 5: Eine CSS-Anweisung zur Definition von Kreisaussehen

Durch dieses Mittel kann SVG auf einfache Weise mit in das gesamte Erscheinungsbild einer Webseite integriert werden und es müssen weniger Attribute in den SVG Elementen selbst spezifiziert werden.

Nun ist eine JavaScript Funktion zu sehen, die dem Dokument neue Kreise hinzufügt. Das nächste Codebeispiel ist ein Ausschnitt von (Dailey, Frost, & Strazzullo, Simple Template,

2017), welches eine komplette Anwendung von SVG mit JavaScript zeigt, bei der Kreise auf Mausklick hinzugefügt und gelöscht werden. Diese ist im Anhang zu finden.

```
function add(evt){
     var C=document.createElementNS(xmlns, "circle")
     C.setAttribute("r", 50)
     C.setAttribute("cx", evt.clientX)
     C.setAttribute("cy", evt.clientY)
     document.documentElement.appendChild(C)
}
```

Code 6: Eine JavaScript Funktion zur Erzeugung von neuen Kreisen

Dabei ist zu beobachten, dass die bekannten Methoden `appendChild` und `setAttribute` benutzt werden, jedoch `createElementNS` zusätzlich einen XML-Namespace als Parameter benötigt, um ein SVG Element zu erzeugen. Auf diese Notwendigkeit wird im Kapitel *Einbettung und Umsetzung* näher eingegangen. Das Objekt `documentElement` ist das Objekt unter dem neue SVG Elemente eingefügt werden können (Dailey, Frost, & Strazzullo, 2012, S. 110).

Nachdem nun die wichtigsten Features von SVG beleuchtet wurden, wird im Folgenden darauf eingegangen, wie SVG in bestehende HTML Anwendungen integriert werden kann.

2.3 Einbettung und Umsetzung

Wie bereits erwähnt, ist SVG vom W3C als offener Standard spezifiziert worden. Daraus resultierend ist es heutzutage laut (W3C SVG Working Group, 2017) auf jedem modernen Browser für Desktop und Mobiles von Haus aus integriert und bedarf somit keinen speziellen Einbindungsaufwand.

Alle SVG Elemente werden, wie jedes andere HTML Element auch, beim Parsen des Dokuments vom User Agent, meist dem Browser, als solche erkannt. Dieser weiß daraufhin, wie er das Element darstellen muss, da er den offiziellen W3C Standard für SVG implementiert hat (W3C SVG Working Group, 2011). Nachteile in der Performance, also der Geschwindigkeit in der die Webseite vollständig dargestellt ist, ergeben sich durch SVG nicht (Dailey, Frost, & Strazzullo, 2012, S. 221).

Um SVG Elemente in einem HTML Dokument zu integrieren, existieren zwei einfache Wege.

Der erste Weg ist, für alle SVG Elemente den umschließenden `<svg>` Tag zu benutzen. Alle weiteren Elemente stehen innerhalb dieses Tags. Der `<svg>` Tag kann wie jedes andere Blockelement im HTML Dokument eingefügt werden (Eisenberg & Bellamy-Royds, 2014, S. 22ff). Bei dieser Vorgehensweise ist jedoch zu beachten, dass CSS und JavaScript innerhalb des `<svg>` Tags über das `<style>` beziehungsweise `<script>` Tag zusätzliche Informationen benötigen, da SVG wie eingangs erwähnt auf XML basiert. Hierzu muss jeweils direkt nach dem öffnenden und direkt vor dem schließenden `<style>` beziehungsweise `<script>` Tag der Eintrag `<![CDATA[` beziehungsweise `]]>` eingefügt werden. Zudem ist es nötig bei Verwendung von JavaScript Funktionen, die SVG Elemente bearbeiten, einen Verweis auf einen sogenannten XML-Namespace anzugeben, aus dem der User Agent die benötigten Informationen bekommt, wie die verwendeten Funktionen angewendet werden sollen (Dailey, Frost, & Strazzullo, 2012, S. 102ff). Dies wird innerhalb des öffnenden `<svg>` Tags zum Beispiel auf folgende Weise durchgeführt:

```
<svg xmlns=”http://www.w3.org/2000/svg”
     xmlns:xlink=”http://www.w3.org/1999/xlink”>
```

Code 7: Definition des XML-Namespaces für SVG

Auf die gleiche Art, wie SVG innerhalb eines HTML Dokuments mithilfe des `<svg>` Tags erstellt wird, kann auch eine eigenständige SVG Datei erstellt werden, die nur den `<svg>` Tag als oberstes Element und dessen Inhalt enthält. Solche Dateien sollten die Endung `.svg` bekommen. Diese Dateien können von modernen Browsern auch direkt angezeigt werden. (W3C SVG Working Group, 2011).

Die zweite Art, wie SVG in HTML integriert werden kann, ist mithilfe des HTML `<img>` Tags. Dabei erhält das `src` Attribut einen Verweis auf eine `.svg` Datei, wodurch diese Datei wie ein ganz normaler `<img>` Tag dargestellt wird (Eisenberg & Bellamy-Royds, 2014, S. 15f).

Es existieren noch andere Arten, wie SVG in HTML integriert werden kann, die unter (W3C SVG Working Group, 2011) aufgelistet sind, zum Beispiel unter Verwendung des `<applet>` Tags, was jedoch im Rahmen dieser Hausarbeit nicht relevant ist und daher unausgeführt bleibt.

Zusammenfassend lässt sich sagen, dass SVG ohne großen Aufwand in HTML Dokumente integriert werden kann.

2.4 Erzeugung

Ein wesentlicher Schritt bei der Verwendung von SVG ist die Art, wie es überhaupt erzeugt wird. Im Folgenden soll geschildert werden, welche Varianten dafür existieren.

Die offensichtlichste Variante ist manuell eine `.svg` Datei oder einen `<svg>` Tag zu erstellen, der die Vektorgrafik enthält, die dargestellt werden soll. Da dies aber gerade bei komplexen Grafiken sehr mühsam bis unüberschaubar komplex werden kann, gibt es einfachere Mittel – schließlich müssen auch keine Rastergrafiken Pixel für Pixel manuell erstellt werden.

SVG kann über programmatische Wege oder mit Designtools einfacher erstellt werden. Beide Wege werden nun kurz vorgestellt und sind aus (Dailey, Frost, & Strazzullo, 2012, S. 191-205) entnommen.

Als Erleichterung der Erzeugung im Bereich der Programmierung, existieren verschiedene Frameworks und Bibliotheken, wie zum Beispiel *JQuery*, *Data Driven Documents* (D3), *Pergola* und *Raphaël*. Beispiele zu diesen Hilfen sind nicht Teil der Arbeit, da diese zu erklären über die Zielsetzung der Arbeit hinausginge. Alle bieten jedoch Methoden oder vordefinierte Wege an, mit denen automatisiert schneller SVG Elemente erzeugt werden können, was somit vor allem bei einer regelmäßigen und logisch aufgebauten Struktur einer Vektorgrafik ein sinnvolles Einsatzgebiet ist.

Eine wesentliche Erleichterung bei der Erzeugung von komplexen Grafiken ist der Einsatz von Designtools, die in der Lage sind völlig ohne Programmieraufwand `.svg` Dateien zu erzeugen. Bei solchen Programmen werden Grafiken oft über Auswahlmenüs erzeugt und in Echtzeit simuliert. Meist bieten diese eine Vielzahl an Möglichkeiten, um Grafiken jeglicher Art zu erzeugen, bei denen diese am Ende als Vektorgrafik gespeichert werden können. Programme, die solche Funktionen besitzen sind unter Anderem weit verbreitet und auch kostenlos nutzbar, wie *Adobe Illustrator* oder *Inkscape*.

In Abbildung 10 ist ein Ausschnitt des Programms *Adobe Illustrator* zu sehen, der verdeutlichen soll, wie damit auf intuitive Art und Weise Grafik erstellt werden kann. Dies soll stellvertretend für alle weiteren Designtools deren Fähigkeiten demonstrieren.

Abbildung 10: Ausschnitt aus dem Programm *Adobe Illustrator*
aus (Adobe Systems, 2017)

Die Erzeugung von SVG mit Designtools ist für komplexe Grafiken, die keine programmierbare Regelmäßigkeit besitzen die einzige sinnvolle Wahl. Auch einfachere Grafiken können unter Umständen auf diese Weise um einiges schneller erzeugt werden, als wenn der Quellcode dafür selbst programmiert wird.

Dieses Kapitel schließt die Übersicht über SVG ab. Es sollte ein Grundverständnis schaffen, um darauf aufbauend im nachfolgenden Kapitel die Einschätzung des Einsatzes von SVG auf der iELP geben zu können.

3 Einsatzmöglichkeiten in der iELP

Zu Beginn dieser Arbeit wurde das Ziel geäußert, eine Einschätzung zur Verwendung von SVG in der iELP zu geben. Mit dem Wissen aus Kapitel 2 ist dies nun möglich. Nachfolgend werden verschiedene Szenarien geschildert, wie solch eine Verwendung aussehen kann.

Da es sich bei SVG um ein Mittel handelt, um Grafiken zu erstellen, ist dessen Einsatzschwerpunkt in der iELP auch bei der Erstellung von fachlichem Inhalt oder anderen Formen von Grafiken, wie Hintergrundbilder, zu sehen. Ein Einsatz darüber hinaus, um es zum Beispiel zur Strukturierung oder zum Layout der Webanwendung zu benutzen, ist nicht zu empfehlen, da dafür vorgesehen Mittel existieren, die diese Aufgabe besser bewältigen.

SVG ist somit eine reine clientseitige Ergänzung der HTML Seiten, die ein Nutzer der iELP auf seinem Browser zu sehen bekommt.

Die Einsatzmöglichkeiten lassen sich in zwei unterschiedliche Bereiche gliedern: Die *Veranschaulichung von statischem Inhalt*, wie Bilder, die in den einzelnen Fachmodulen der Webanwendung zu finden sind; sowie die *Animation von Inhalt*, die interaktiv oder selbständig verändert und ergänzt werden kann. Beide Möglichkeiten werden im Folgenden näher angesprochen.

3.1 Veranschaulichung von statischem Inhalt

In der iELP existieren bisher einige Rastergrafiken, die dazu dienen, Fachwissen zu vermitteln und darzustellen. Eine dieser Rastergrafiken ist in der untenstehenden Abbildung zu sehen.

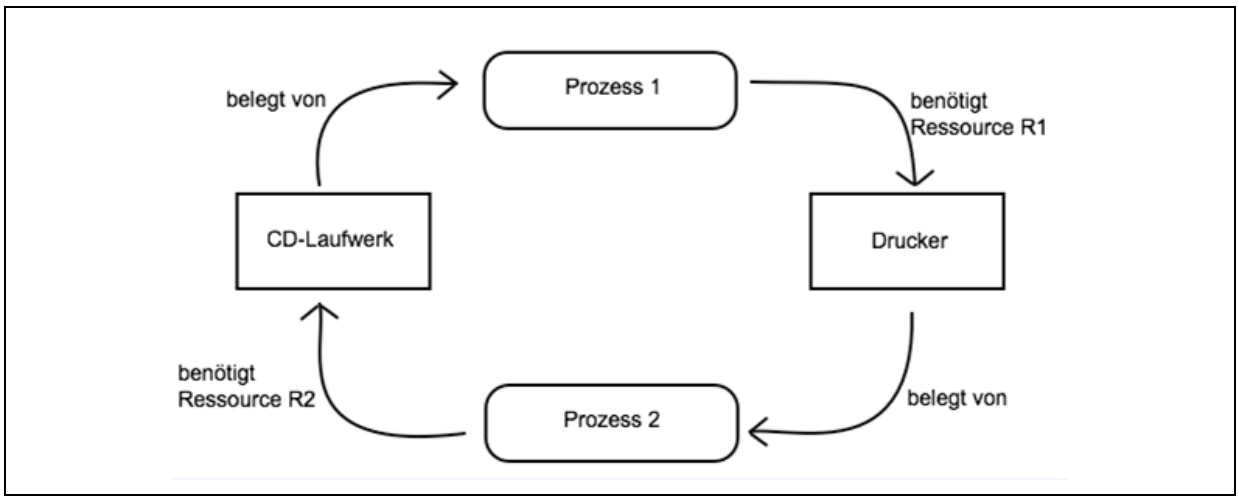

Abbildung 11: Eine Rastergrafik der iELP
aus (Hochschule Coburg, 2016)

Der in Kapitel 2.1 angesprochene Nachteil von Rastergrafiken gilt somit auch für diese Bilder. An dieser Stelle kann mit SVG angesetzt werden, indem alle Rastergrafiken neu erstellt werden und als Vektorgrafik in einer eigenen `.svg` Datei zur Verfügung stehen. Wenn für die iELP neue Bilder erstellt werden, sollten diese grundsätzlich als `.svg` Datei erstellt werden, anstatt Rastergrafiken wie `.jpg` oder `.png` Dateien zu verwenden.

Die bisherige Einbindung der Rastergrafiken erfolgt über das HTML `<img>` Tag. Wie in Kapitel 2.3 geschildert, können `.svg` Dateien auf die gleiche Art eingefügt werden, was den Integrationsaufwand für diese Verbesserung sehr gering macht.

Neben dem Ersetzen von bisherigen Rastergrafiken können neue Bilder auch beliebig komplex konstruiert werden, da sie, wie jede Vektorgrafik, dadurch keinen Qualitätsverlust durch Skalierung erleiden. Dies könnte je nach Bedarf an neuen Bildern ebenfalls einen Mehrwert bieten. Es können zudem auch Bilder, die kein Wissen vermitteln, sondern die Webanwendung an sich gestalten sollen, wie Hintergrundbilder, über SVG erstellt werden.

Generell kann mit der Verwendung von SVG also der Qualitätsverlust von Bildern jeglicher Form und Verwendung auf der iELP vermieden werden.

3.2 Animation von Inhalt

Neben statischen Grafiken existieren in der iELP auch dynamisch erzeugte oder interaktive Teile, wie sogenannte *Simulationen* oder *Aufgaben.* Bei diesen wird auf der HTML Seite nach Eingaben des Benutzers häufig neuer Inhalt hinzugefügt. In nachfolgender Abbildung ist ein Beispiel einer Simulation zu sehen, die den generellen Aufbau solcher Anwendungen innerhalb der iELP verdeutlichen soll.

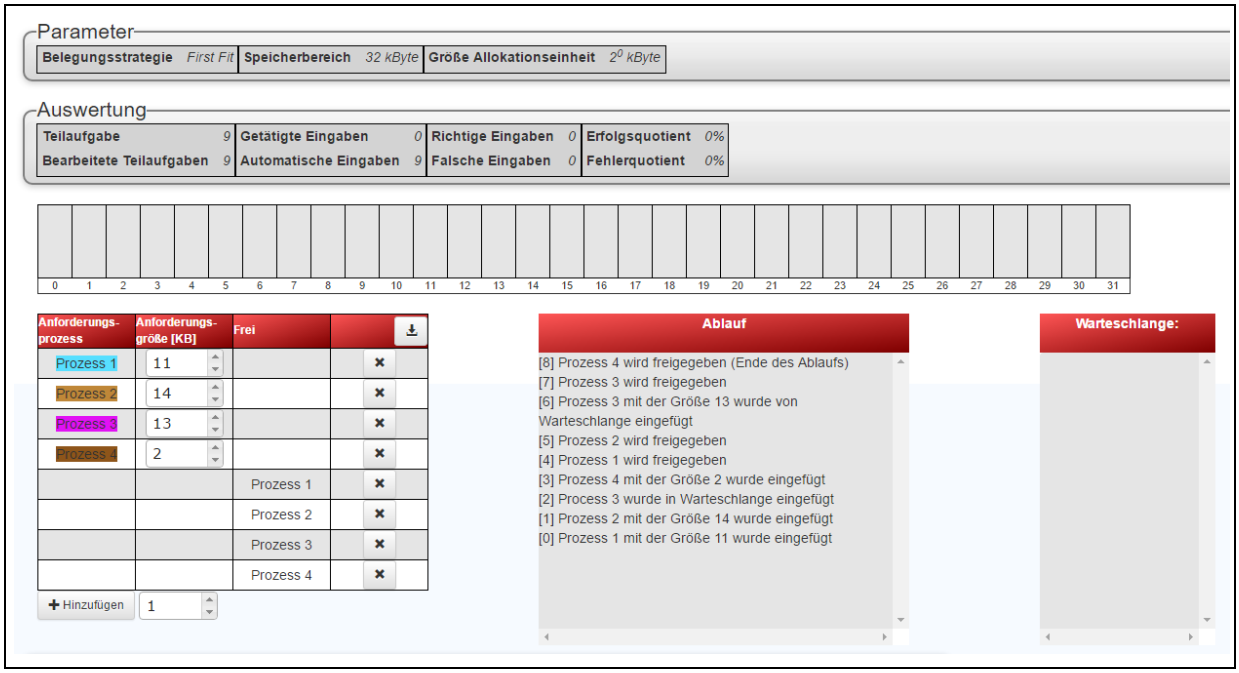

Abbildung 12: Ausschnitt einer Simulation der iELP
aus (Hochschule Coburg, 2016)

Dabei wird ein bestimmtes Fachwissen interaktiv vermittelt, welches zur Darstellung lediglich auf textbasierten Inhalt zurückgreift. Dieser Inhalt ist skalierbar und erleidet keinen Qualitätsverlust beim Vergrößern. Eine Möglichkeit an dieser Stelle SVG zu integrieren ist fachlich zu beurteilen. Es kann in Betracht gezogen werden, an dieser Stelle SVG Grafiken zur Steigerung der Veranschaulichung zu ergänzen. Da SVG, wie in Kapitel 2.2.3 geschildert, über JavaScript manipuliert werden kann, ist es möglich, dies ebenfalls interaktiv zu gestalten.

Eine komplett neue Möglichkeit wird mit SVG zudem geschaffen: Die in Kapitel 2.2.2 geschilderten Animationen und Filter können benutzt werden, um Bewegung als neue Form der Wissensvermittlung zu gebrauchen. Bei bestimmten Sachverhalten kann es sinnvoll sein, eine Bewegung darzustellen, beispielsweise ein Graph oder eine Schwingung, die immer wieder auftritt. Dies ist mit SVG möglich und kann daher für die iELP genutzt werden.

Nachdem die Einsatzmöglichkeiten von SVG in der iELP nun geschildert wurden, wird im folgenden Fazit nun ein Abschluss dafür gefunden.

4 Fazit

Eingangs wurde die Zielsetzung der Arbeit um vier Fragestellungen erweitert, die diese genauer spezifiziert haben. Mit dem Wissen aus Kapitel 2 und 3 ist es nun möglich, diese zu beantworten um letztendlich eine Einschätzung über SVG in der iELP geben zu können.

Nachfolgend werden alle vier Fragestellungen aufgegriffen und kommentiert.

1) *Kann SVG die Nutzerfreundlichkeit durch seine Features steigern?*
   Diese Frage ist ganz klar mit einem *ja* zu beantworten. Der bisherige Einsatz von Rastergrafiken (s. Kapitel 3.1) ist nicht gerade nutzerfreundlich. Wird die iELP beispielsweise auf einem mobilen Gerät mit kleinerem Display verwendet oder vergrößert der Nutzer die Ansicht auf die Seite, erleiden diese Grafiken einen Qualitätsverlust (s. Kapitel 2.1). Zwar existieren auch Wege, dieses Verhalten mit Rastergrafiken zu lösen, beispielsweise durch Verwendung von unterschiedlich großen Bildern je nach Größe der Seite, jedoch bietet SVG hierfür eine endgültige und einfachere Lösung.
2) *Kann SVG Veranschaulichungen darstellen und in welchem Maße vorhandene ersetzen?*
   Wie aus Kapitel 2 hervorgeht, ist SVG generell dafür gedacht, grafische Veranschaulichungen zu erstellen. In der iELP bereits vorhandene Grafiken können, wie in Kapitel 3.1 beschrieben, ersetzt werden.
3) *Kann SVG animierten Inhalt sinnvoll gestalten?*
   In den Kapiteln 2.2.2 und 2.2.3 wird ausführlich erwähnt, auf welche Art Animationen und Interaktionen mit SVG möglich sind. In der iELP kann dies, wie in Kapitel 3.2 beschrieben, auf sinnvolle Weise umgesetzt werden.
4) *Kann SVG durch passende Entwicklungswerkzeuge einfach entworfen werden?*
   Diese Frage ist ebenfalls mit einem klaren *ja* zu beantworten. Wie in Kapitel 2.4 geschildert, kann SVG durch Designtools entworfen werden. Dies ermöglicht es, die Erstellung von Grafiken auch in Bereiche auszulagern, in denen keine Programmierkenntnisse vorliegen. Somit ist SVG hochflexibel erstellbar. Soll es doch manuell programmiert werden, gibt es dafür, wie erwähnt, auch erleichternde Werkzeuge.

Abschließend lässt sich der Einsatz von SVG in der iELP generell empfehlen. Es bietet viele Vorteile und ist ohne viel Aufwand oder Verlusten bei der Performance (s. Kapitel 2.3) in die

bestehende Webanwendung integrierbar. Zwar sind Rastergrafiken generell verbreiteter als Vektorgrafiken, doch wurden die Unterschiede der zwei Technologien beleuchtet, aus denen ein klarer Vorteil der Vektorgrafiken entsprungen ist. Der Einsatz von SVG ist jedoch mehr von fachlichen Anforderungen abhängig. Da diese wohl vorrangig zur Präsentation von Wissen benutzt werden könnten, hängt es sehr davon ab, wie groß der Bedarf an zusätzlichen Bildern ist und ob sich der Aufwand lohnt, die vorhandenen Bilder zu ersetzen. Ob zudem der Gebrauch von SVG bei Animationen sinnvoll ist, hängt ebenfalls von dem Mehrwert an Wissensvermittlung, der dadurch entsteht ab. Dies ist von zuständigen Personen abzuwägen und zu beurteilen.

Damit schließt die Evaluation in dieser Hausarbeit ab. Nachfolgend wird noch kurz umrissen, wie der weitere Vorgang im Kontext dieser Arbeit aussehen wird.

5 Ausblick

Die Aufgabe eine Einschätzung zur Verwendung von SVG in der iELP zu geben ist mit dieser Arbeit abgeschlossen. Wie eingangs erwähnt wurde, ist diese Arbeit innerhalb eines Projektteams entstanden, das verschiedene Verbesserungsmöglichkeiten untersuchen soll. Es ist offensichtlich, dass davon nicht jede Verbesserungsmöglichkeit sofort eingesetzt werden kann oder dass dies überhaupt sinnvoll ist. Daher ist der nächste Schritt, nun zu beurteilen, welche der Technologien in welchem Maße eingesetzt wird und es bleibt zu planen, in welchem Zeitraum dies geschieht. Diese Arbeit wird dabei die Grundlage sein, um die Beurteilung für die Verbesserungsmöglichkeit *Scalable Vector Graphics* durchzuführen.

Literaturverzeichnis

(Adobe Systems, 2017)	Adobe Systems: Paul Douarad AI Mask, http://blogs.adobe.com/adobeillustrator/files/2016/01/Paul_Douarad_AI_Mask_sm.png (Zugriff: 05.01.2017)
(Dailey, Frost, & Strazzullo, 2012)	Dailey, D.; Frost, J.; Strazzullo, D.: Building Web Applications with SVG, 1. Auflage, O'Reilly, California, 2012.
(Dailey, Frost, & Strazzullo, Simple Template, 2017)	Dailey, D.; Frost, J.; Strazzullo, D.: Simple Template, http://srufaculty.sru.edu/david.dailey/svg/simpleTemplate.svg (Zugriff: 05.01.2017)
(Eisenberg & Bellamy-Royds, 2014)	Eisenberg, J.; Bellamy-Royds, A.: SVG Essentials, 2.Auflage, O'Reilly, USA, 2014.
(Hochschule Coburg, 2016)	Hochschule Coburg: Interaktive E-Learning Plattform, http://10.6.4.250:8080/ielm_portal/ (Zugriff 05.01.2017)
(TheCraftChop, 2016)	TheCraftChop: darth_1.svg, http://thecraftchop.com/files/images/darth_1.svg (Zugriff 05.01.2017)
(W3C DOM IG, 2009)	W3C DOM IG: Document Object Model (DOM), https://www.w3.org/DOM/ (Zugriff 05.01.2017)
(W3C SVG Working Group, 2011)	W3C SVG Working Group: Scalable Vector Graphics (SVG) 1.1 (Second Edition) W3C Recommendation, https://www.w3.org/TR/SVG11/ (Zugriff: 05.01.2017)

(W3C SVG Working Group, 2017) W3C SVG Working Group: Scalable Vector Graphics (SVG), https://www.w3.org/Graphics/SVG/ (Zugriff: 05.01.2017)

(Wikipedia, 2016) Wikipedia: Antialiasing (Computergrafik), https://de.wikipedia.org/wiki/Antialiasing_(Computergrafik) (Zugriff: 05.01.2017)

Glossar

Antialiasing	Technik, um eckige Abschnitte in einem verpixelten Bild zu retuschieren. Dabei werden unter anderem durch entsprechende Farbgebung umliegender Pixel Ecken schwächer dargestellt (Wikipedia, 2016).
Attribut	Wie aus HTML bekannt, ergänzt es ein Element um Eigenschaften, wie zum Beispiel die `Id:` `<p id="name">`
Blockelement	Ein Element, das immer in einer neuen Zeile des HTML Dokuments beginnt und diese voll ausfüllt.
Document Object Model	(W3C DOM IG, 2009): *„The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents."*
Element	Wie aus HTML bekannt, ist ein Element die Definition eines neuen Abschnittes in der Datei. Zum Beispiel `<p>`, `<svg>` oder `<circle>`. Charakteristisch sind die Symbole `<` und `>` bzw. `/>` um jedes Element platziert.
Geometrische Figur	Ein mathematisch eindeutig beschriebenes Objekt im hier zweidimensionalen Raum mit einer bestimmten Form. Zum Beispiel ein Kreis, Rechteck, Linie, usw.
Grafische Information	Eine Information, die visualisiert werden kann. Zum Beispiel die Information darüber, wie ein Kreis aussehen soll.
Nutzerfreundlichkeit	Das subjektive Empfinden, wie komfortabel und einfach sich etwas (hier: eine Webseite) überblicken und benutzen lässt.
Skalierbarkeit	Hier: Die Fähigkeit Bilder in ihrer Größe zu verändern, ohne dabei einen Qualitätsverlust zu erleiden.
Tag	Ein Synonym für Element.

User Agent	Die Instanz, die das Parsen der HTML Dokumente durchführt. Meist ein Browser.
Verpixeln	Das deutliche Hervortreten von einzelnen Bildpunkten, sogenannten Pixeln, in einem Bild. Das Bild wird dann oft als eckig oder unscharf wahrgenommen.

Anhang A 1. Vollständige Codebeispiele

Alle Codebeispiele lassen sich benutzen, indem sie in eine Textdatei der Endung `.svg` als einziger Inhalt eingefügt werden. Sie können mit einem modernen Browser angezeigt werden.

Codebeispiel 1

Eigenentwicklung

```
<svg width=”100%” height=”100%”>
  <circle cx="50" cy="50" r="40" stroke="black" stroke-
          width="3" fill="red" />
</svg>
```

Codebeispiel 2

Eigenentwicklung

```
<svg width=”100%” height=”100%”>
  <path d="M 100 200 q 150 -300 300 0" stroke="blue" stroke-
        width="5" fill="none" />
</svg>
```

Codebeispiel 3

Eigenentwicklung

```
<svg width=”100%” height=”100%”>
  <circle cx="50" cy="50" r="50" fill="red">
     <animateMotion dur="5s"
                    values="0,0; 500,0; 250,250; 0,0"
                    repeatCount="indefinite" />
  </circle>
</svg>
```

Codebeispiel 4

Eigenentwicklung

```
<svg width=”100%” height=”100%”>
  <filter id="name">
    <feGaussianBlur stdDeviation="5" />
  </filter>
  <circle cx="100" cy="100" r="50" fill="green"
          filter="url(#name)" />
</svg>
```

Codebeispiel 5

Eigenentwicklung, hier die Endung `.html` für Textdateien benutzen.

```
<html>
<head>
 <style>
   circle{
     fill: blue;
     stroke-width: 5;
   }
 </style>
</head>
<body>
 <svg width=”100%” height=”100%”>
  <circle cx="50" cy="50" r="20"/>
  <circle cx="150" cy="50" r="20"/>
  <circle cx="250" cy="50" r="20"/>
 </svg>
</body>
</html>
```

Codebeispiel 6

Aus (Dailey, Frost, & Strazzullo, Simple Template, 2017) entnommen.

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%"
xmlns:xlink="http://www.w3.org/1999/xlink"
>
<script><![CDATA[
xmlns="http://www.w3.org/2000/svg"
xlink="http://www.w3.org/1999/xlink"
document.documentElement.setAttribute("onclick","removeIt(evt)
")
//note: previous example named the function "remove()". This
collides with a new
// .remove() method defined in the Working Draft for DOM 4:
http://www.w3.org/TR/domcore/
function removeIt(evt){
     if (evt.target.nodeName=="rect") add(evt)
```

```
     else document.documentElement.removeChild(evt.target)
}
function add(evt){
     var C=document.createElementNS(xmlns,"circle")
     C.setAttributeNS(null, "r", 50)
     C.setAttributeNS(null, "cx", evt.clientX)
     C.setAttributeNS(null, "cy", evt.clientY)
     document.documentElement.appendChild(C)
}
]]>
</script>
<rect width="100%" height="100%" fill="white"/>
<circle r="50"/>
<text font-size="12"  x="50" y="20" >Click something to remove
it</text>
<text font-size="12"  x="50" y="80" >Click nothing to add
something</text>
</svg>
```

Codebeispiel 7

Vergleiche dazu die Definition des `<svg>` Tags von Codebeispiel 6.

Abbildung 5

Aus (TheCraftChop, 2016) entnommen.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 16.0.0, SVG Export Plug-In .
SVG Version: 6.00 Build 0)  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd" [
     <!ENTITY ns_flows "http://ns.adobe.com/Flows/1.0/">
     <!ENTITY ns_extend
"http://ns.adobe.com/Extensibility/1.0/">
     <!ENTITY ns_ai
"http://ns.adobe.com/AdobeIllustrator/10.0/">
     <!ENTITY ns_graphs "http://ns.adobe.com/Graphs/1.0/">
]>
<svg version="1.1" id="Layer_1" xmlns:x="&ns_extend;"
xmlns:i="&ns_ai;" xmlns:graph="&ns_graphs;"
```

```
      xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:a="http://ns.adobe.com/AdobeSVGViewerExtensions/3.0/"

      x="0px" y="0px" width="864px" height="864px" viewBox="0 0
864 864" enable-background="new 0 0 864 864"
xml:space="preserve">

<path fill="#FFFFFF" d="M458.25,792.5c-8.5,0-17,0-25.5,0c-
21.112-5.83-42.296-11.246-64.361-11.985

     c-58.842-1.974-117.704-3.528-176.517-6.159c-32.104-1.438-
63.659-7.021-93.816-18.846c-15.44-6.053-30.89-12.103-35.806-
30.51

     c0-6.5,0-13,0-19.5c1.256-4.749,2.329-9.555,3.796-
14.237c18.597-59.358,38.842-118.186,67.195-173.654

     c28.775-56.295,45.75-114.58,41.118-178.574c-3.335-
46.081,8.048-89.752,27.412-131.353c21.35-45.864,53.91-
80.509,101.053-100.818

     c31.382-13.518,63.825-22.991,97.487-28.366c14.756-
2.356,29.623-4.02,44.439-5.998c11,0,22,0,33,0

     c1.418,0.41,2.812,1.056,4.256,1.195c25.911,2.512,50.355,10
.103,73.57,21.637c41.109,20.425,71.354,50.689,85.787,95.262

     c5.689,17.57,11.737,35.03,17.954,52.421c7.361,20.596,15.84
2,40.604,34.241,54.253c11.873,8.809,19.508,20.653,24.281,34.35
3

     c16.578,47.589,34.249,94.85,49.018,142.998c12.193,39.752,2
1.037,80.543,30.951,120.974c1.882,7.681,2.531,15.838,2.51,23.7
71

     c-0.039,15.029-9.988,24.733-20.582,33.232c-30.249,24.262-
62.305,45.94-96.346,64.556c-17.572,9.609-35.586,18.881-
56.123,19.295

     c-15.839,0.32-27.046,7.241-35.737,19.701c-1.997,2.861-
4.3,5.518-6.564,8.18c-6.673,7.843-14.681,13.51-25.135,15.043

     c-19.746,2.893-39.468,5.975-
59.253,8.58C486.514,789.805,472.362,791.008,458.25,792.5z"/>

<g>

     <path d="M459.25,793.5c-8.5,0-17,0-25.5,0c-21.112-5.83-
42.296-11.246-64.361-11.985c-58.842-1.974-117.704-3.528-
176.517-6.159

          c-32.104-1.438-63.659-7.021-93.816-18.846c-15.44-
6.053-30.89-12.103-35.806-30.51c0-6.5,0-13,0-19.5

          c1.256-4.749,2.329-9.555,3.796-14.237c18.597-
59.358,38.842-118.186,67.195-173.654c28.775-56.295,45.75-
114.58,41.118-178.574
```

c-3.335-46.081,8.048-89.752,27.412-131.353c21.35-45.864,53.91-80.509,101.053-100.818c31.382-13.518,63.825-22.991,97.487-28.366

c14.756-2.356,29.623-4.02,44.439-5.998c11,0,22,0,33,0c1.418,0.41,2.812,1.056,4.256,1.195

c25.911,2.512,50.355,10.103,73.57,21.637c41.109,20.425,71.354,50.689,85.787,95.262c5.689,17.57,11.737,35.03,17.954,52.421

c7.361,20.596,15.842,40.604,34.241,54.253c11.873,8.809,19.508,20.653,24.281,34.353c16.578,47.589,34.249,94.85,49.018,142.998

c12.193,39.752,21.037,80.543,30.951,120.974c1.882,7.681,2.531,15.838,2.51,23.771c-0.039,15.029-9.988,24.733-20.582,33.232

c-30.249,24.262-62.305,45.94-96.346,64.556c-17.572,9.609-35.586,18.881-56.123,19.295c-15.839,0.32-27.046,7.241-35.737,19.701

c-1.997,2.861-4.3,5.518-6.564,8.18c-6.673,7.843-14.681,13.51-25.135,15.043c-19.746,2.893-39.468,5.975-59.253,8.58

C487.514,790.805,473.362,792.008,459.25,793.5z M478.454,633.269c-1.434-3.416-2.427-5.641-3.306-7.909

c-1.864-4.804-1.963-7.872,4.935-8.137c6.12-0.235,12.255-1.975,18.229-3.614c6.467-1.775,12.644-5.567,19.142-6.098

c21.604-1.764,43.301-3.431,64.952-3.319c28.204,0.146,56.4,2.005,84.597,3.201c2.459,0.104,4.951,0.561,7.33,1.207

c5.069,1.379,7.625,4.584,4.127,9.305c-5.875,7.933-11.055,17.257-18.828,22.725c-16.475,11.587-34.575,20.837-51.731,31.491

c-11.813,7.336-23.771,14.669-34.535,23.395c-9.718,7.876-16.515,8.702-25.142-0.442c-2.215-2.346-4.516-4.846-5.906-7.699

c-7.202-14.784-14.061-29.735-21.176-44.562c-2.473-5.152-4.881-10.437-8.146-15.078c-3.617-5.142-8.719-5.481-13.71-1.574

c-4.507,3.529-8.437,7.179-3.627,13.535c7.827,10.346,15.146,21.077,23.027,31.378c10.469,13.681,21.534,26.91,31.797,40.739

```
     c3.81,5.135,5.812,11.572,9.103,17.14c4.055,6.862,7.229,7.62,12.919,2.332c5.434-5.05,9.962-11.07,14.904-16.651

          c3.755-4.241,6.845-9.506,11.414-12.523c37.899-25.037,76.103-49.611,114.137-74.445c6.618-4.32,12.87-9.2,19.664-14.09

          c-2.161-3.102-3.618-5.079-4.956-7.136c-5.714-8.783-12.105-17.216-16.945-26.46c-12.4-23.688-14.076-47.115,3.727-69.439

          c3.244-4.068,5.823-8.721,8.324-13.314c7.492-13.772,6.818-26.73-4.199-38.588c-3.059-3.292-5.781-6.893-8.671-10.342

          c-8.528-10.178-11.622-21.866-9.787-34.952c1.041-7.419,2.046-14.844,3.011-22.273c2.164-16.678-0.443-31.92-13.469-43.936

          c-19.312-17.815-50.833-21.307-73.152-7.632c-5.506,3.374-10.051,8.317-15.036,12.539c0.509,0.758,1.018,1.515,1.526,2.273

          c9.943-3.165,19.706-7.258,29.88-9.229c9.854-1.909,20.424-3.462,30.159-1.911c16.328,2.602,31.372,23.596,33.252,43

          c1.484,15.315-2.539,23.404-16.313,30.484c-14.564,7.486-30.354,9.855-46.55,8.729c-10.981-0.763-22.229-1.957-30.531-10.381

          c-6.683-6.78-14.402-9.226-23.688-8.699c-4.41,0.25-8.894-0.775-14.354-1.324c5.979-9.889,18.774-1.801,23.352-12.43

          c-13.718-6.797-26.897-2.997-40.413-0.481c-0.796,9.265,3.559,15.822,8.742,22.025c4.321,5.171,9.071,10.029,12.931,15.519

          c1.646,2.341,2.532,6.13,2,8.927c-2.135,11.221-5.416,22.229-7.424,33.467c-2.978,16.668-2.671,16.725-19.336,20.951

          c-1.212,0.307-2.439,0.545-3.645,0.872c-37.841,10.272-70.521,27.949-90.891,63.317c-0.122,0.212-0.818,0.095-0.247,0.046

          c-4.236-14.297-8.217-28.416-12.625-42.402c-3.67-11.649-7.744-23.177-11.841-34.688c-1.618-4.547-0.47-8.005,3.511-10.216

          c7.599-4.222,14.999-9.299,23.157-11.924c25.544-8.221,51.871-9.776,78.517-6.918c4.694,0.504,9.556,0.866,14.188,0.217

          c7.875-1.106,9.75-6.14,4.367-11.821c-4.257-4.492-9.312-8.249-14.152-12.16c-8.39-6.781-15.646-8.579-26.99-2.574
```

```
          c-38.679,20.476-100.713,10.76-132.82-18.967c-3.724-
3.448-6.777-7.621-10.399-11.751c7.824-6.867,14.683-
13.042,21.707-19.024

          c25.693-21.88,54.406-34.97,89.192-
31.508c9.396,0.935,18.978,0.255,28.461-0.056c3.464-
0.114,6.892-1.312,10.336-2.015

          c-4.06-3.466-8.05-5.422-12.217-6.865c-50.346-17.43-
118.26,7.616-145.564,53.474c-2.785,4.678-6.199,9.034-
9.74,13.182

          c-1.3,1.523-4.302,3.292-5.697,2.762c-9.955-3.779-
20.314-7.23-26.854-18.703c9.325-9.894,18.046-20.589,28.215-
29.663

          c29.278-26.126,61.776-46.324,102.568-47.13c28.562-
0.565,56.77,2.7,80.033,21.578c17.647,14.322,37.112,16.594,57.8
83,10.926

          c18.51-5.051,36.665-11.41,54.938-17.315c23.836-
7.704,43.664-
2.679,58.668,18.119c0.875,1.213,1.918,2.303,2.855,3.472


     c18.217,22.709,29.039,49.149,35.375,76.997c13.398,58.865,2
5.358,118.059,37.795,177.141c2.428,11.527,4.342,23.163,6.565,3
5.143

          c10.409-6.387,13.798-14.273,10.567-27.893c-13.355-
56.315-26.619-112.663-40.963-168.732

          c-8.303-32.457-19.559-64.033-38.03-92.424c-8.051-
12.375-16.534-24.492-30.646-30.965c-7.486-3.434-12.738-8.762-
16.349-16.44

          c-3.066-6.521-8.722-10.684-18.978-
9.412c3.138,3.509,5.438,5.773,7.349,8.329c1.209,1.617,1.836,3.
668,2.725,5.525

          c-2.267,0.444-4.687,1.639-6.772,1.21c-12.941-2.666-
21.201-16.271-18.541-
31.767c11.372,0.938,22.896,1.887,36.106,2.977

          c-6.739-10.31-14.737-13.573-24.15-14.089c-18.273-
1.004-34.93,5.295-51.272,12.181c-6.87,2.896-10.083,0.887-
12.708-5.191

          c-2.062-4.773-4.637-9.324-6.983-13.975c-0.747,0.271-
1.494,0.54-2.241,0.81c0,9.656-0.414,19.335,0.152,28.958

          c0.35,5.958-2.189,8.031-7.426,8.543c-5.22,0.51-
10.418,1.579-15.638,1.652c-6.805,0.096-8.489,3.328-7.824,9.545

          c0.54,5.05,0.11,10.204,0.11,16.643c-8.292-3.509-
14.893-6.326-21.512-9.099c-17.034-7.135-34.123-14.139-51.089-
21.432
```

```
          c-6.269-2.695-11.876-6.47-15.383-12.41c0.355-
0.848,0.71-1.696,1.066-
2.543c26.603,5.76,53.204,11.52,80.299,17.385

          c0.551-3.886,1.001-5.821,1.075-7.769c0.961-25.295-
0.156-50.441-5.435-75.295c-8.645-40.701-30.447-71.676-67.396-
91.539

          c-12.536-6.739-25.917-12.364-37.262-20.736c-12.56-
9.268-23.964-8.419-36.394-1.67c-6.36,3.454-12.617,7.098-
19.657,11.072


     c14.944,10.7,18.098,13.289,25.358,28.521c9.163,19.225,21.6
07,35.729,36.405,50.901c16.741,17.165,32.707,35.087,48.961,52.
726

          c6.024,6.538,11.91,13.204,18.075,20.049c0.617-
1.188,1.054-1.627,0.984-1.961c-0.406-1.951-0.86-3.899-1.428-
5.81

          c-5.271-17.752-15.031-33.038-25.754-47.915c-11.088-
15.383-21.906-30.985-32.224-46.893

          c-9.937-15.322-19.255-30.943-34.484-41.956c-3.369-
2.437-5.483-6.609-8.175-9.981c0.349-0.667,0.697-1.335,1.045-
2.002

          c4.805,0,10.129-
1.33,14.298,0.297c7.057,2.755,13.466,7.248,19.994,11.257c6.132
,3.767,11.612,9.967,18.181,11.38


     c15.489,3.334,25.324,13.104,32.488,25.597c19.205,33.486,33
.6,69.082,42.99,106.553c1.827,7.289,1.411,15.14,2.174,24.512

          c-17.334-2.682-32.862-4.732-48.25-7.546c-25.035-
4.577-49.833-11.186-75.57-9.649c-9.496,0.567-18.892,2.829-
30.121,4.6


     c3.654,6.9,5.977,11.78,8.733,16.401c13.11,21.975,10.621,29
.093-12.405,40.624c-9.918,4.966-19.76,10.936-27.962,18.311

          c-19.381,17.43-34.821,38.39-43.238,63.092c-
10.121,29.704-19.105,59.9-26.576,90.375c-8.066,32.911-
14.202,66.317-20.376,99.657

          c-2.28,12.316-5.47,17.602-17.784,20.641c-3.865,0.954-
7.837,1.637-11.802,1.993c-18.864,1.694-36.735,5.71-
52.894,16.696

          c-19.185,13.045-39.299,24.713-58.743,37.391c-
6.85,4.466-13.585,9.508-19.169,15.423c-8.75,9.268-
6.374,19.433,5.466,25.765

          c1.199-1.377,2.421-2.973,3.834-4.375c32.764-
32.524,69.547-58.391,115.186-69.656c6.048-1.493,11.941-
3.608,17.965-5.212
```

```
          c18.231-4.852,29.205-16.6,32.991-34.935c4.137-
20.032,7.788-40.166,11.994-60.183c2.382-11.335,5.452-
22.526,8.56-35.199

          c55.09,33.814,110.098,62.884,168.714,85.382c-
9.04,0.99-17.999,1.756-26.786,0.853c-30.549-3.14-61.025-6.98-
91.558-10.289

          c-10.656-1.155-21.39-1.96-32.099-2.098c-5.187-0.067-
10.918,1.084-
11.914,8.921c51.008,8.378,101.618,15.94,153.43,18.369

          c-5.063,3.899-10.309,7.206-15.981,8.301c-
18.129,3.495-36.413,6.193-54.643,9.157c-21.184,3.443-
42.491,6.336-62.07,16.048

          c-13.052,6.474-21.492,16.339-22.377,31.234c-
0.829,13.947-0.868,27.941-1.312,41.914c-0.454,14.268-
4.207,27.079-15.908,36.513

          c-2.352,1.897-4.166,4.456-
6.228,6.71c0.466,0.783,0.932,1.564,1.397,2.346c6.969-
2.305,14.401-3.727,20.812-7.093

          c14.189-7.45,19.966-21.481,25.292-
37.505c2.578,4.43,4.737,7.16,5.869,10.266c3.757,10.301,7.462,2
0.643,10.532,31.16

          c1.464,5.014,4.23,6.666,7.905,3.775c5.143-
4.045,9.584-9.05,13.974-13.959c3.517-3.933,3.165-7.342-0.22-
12.262

          c-15.516-22.549-19.417-46.784-6.891-72.074c1.734-
3.502,5.962-7.447,9.577-8.076c32.454-5.644,65.027-
10.602,97.57-15.734

          C460.482,635.955,469.014,634.702,478.454,633.269z
M615.702,173.453c0.593-0.636,1.186-1.271,1.778-1.907

          c-10.008-54.612-138.644-113.155-192.815-
86.213c15.367,5.558,29.351,11.541,43.887,15.488c7.417,2.015,15
.878-0.196,23.778,0.497


     c22.216,1.949,41.157,12.959,60.587,22.61c8.022,3.985,17.54
8,5.876,24.271,11.331c10.526,8.54,19.212,19.339,28.766,29.095

          C609.061,167.527,612.445,170.427,615.702,173.453z
M463.371,369.152c-12.226,0.174-19.637,6.572-19.549,16.878

          c0.081,9.549,5.773,15.878,14.229,15.819c10.388-
0.072,20.769-9.312,20.381-
18.141C478.09,375.898,470.997,369.044,463.371,369.152

          z
M557.779,358.047c10.773,0.829,21.032,1.618,30.803,2.371c2.549-
7.556,0.018-10.603-5.773-9.954
```

```
          c-12.088,1.356-24.127,3.193-36.125,5.192c-1.204,0.2-
2.879,2.895-2.828,4.38c0.218,6.413,0.996,12.806,1.636,19.942

          c14.802-1.483,29.373,1.521,42.479-6.832c-2.491-1.479-
4.991-2.266-7.543-2.497c-5.463-0.495-10.992-0.399-16.424-1.098

          C555.371,368.443,554.997,367.535,557.779,358.047z
M629.182,405.464c9.703-0.083,19.029-8.245,18.992-16.622

          c-0.034-8.016-6.921-15.264-14.34-15.094c-11.839,0.27-
18.963,6.642-18.743,16.764

          C615.296,399.921,620.588,405.536,629.182,405.464z"/>

     <path fill-rule="evenodd" clip-rule="evenodd"
d="M377.782,473.36c0.68,14.806,1.486,27.765,1.82,40.737


     c0.43,16.722,7.633,29.386,22.438,37.037c12.17,6.288,24.87,
11.544,37.237,17.463c7.575,3.626,14.964,7.641,22.436,11.479

          c-0.42,0.903-0.84,1.807-1.26,2.71c-16.094-6.944-
32.842-12.693-48.152-21.071c-36.759-20.116-72.839-41.474-
109.203-62.313

          c-12.053-6.906-12.806-7.021-10.208-20.878c3.387-
18.073,7.272-36.145,12.519-53.747c3.485-11.69,8.557-
12.852,20.056-7.993


     c2.749,1.161,5.577,2.359,7.975,4.085c21.064,15.158,42.585,
29.752,62.896,45.869c22.982,18.236,37.385,41.953,39.558,71.998

          c0.07,0.959,0.01,1.929,0.01,2.894c-0.666,0.356-
1.33,0.714-1.995,1.07c-2.637-4.399-6.351-8.487-7.734-13.251

          C418.715,503.758,397.886,489.993,377.782,473.36z"/>

     <path fill-rule="evenodd" clip-rule="evenodd"
d="M686.37,583.229c-4.685,3.147-8.782,5.9-14.499,9.742

          c-9.465-10.369-19.078-20.899-29.857-
32.707c0,9.007,0,17.078,0,26.272c-6.311,0-11.257,0.417-16.062-
0.225

          c-1.704-0.228-3.985-2.813-4.43-4.694c-2.805-11.861-
5.05-23.854-7.734-35.746c-1.031-4.57-2.734-8.99-4.131-13.478

          c-1.312,0.153-2.626,0.306-3.938,0.459c-0.132,6.106-
0.816,12.261-
0.269,18.306c0.854,9.419,2.869,18.735,3.645,28.157

          c0.187,2.271-2.72,6.9-3.936,6.812c-4.795-0.354-
11.949,4.175-13.93-3.885c-2.553-10.391-3.619-21.141-6.068-
31.562

          c-1.704-7.254-4.656-14.216-8.332-20.984c-
11.102,18.997-1.317,37.432,0.044,55.562c-0.577,0.538-
0.74,0.828-0.945,0.861
```

```
          c-14.9,2.423-14.908,2.423-18.466-12.238c-2.924-
12.055-5.825-24.114-8.939-37.01c-6.734,3.808-8.496,9.405-
8.077,15.475


     c0.667,9.694,2.129,19.332,3.188,29.001c0.264,2.41,0.271,4.
851,0.421,7.709c-5.145,0.823-9.691,1.552-14.521,2.325

          c-1.612-8.774-3.08-16.767-4.463-24.294c-8.951,6.96-
17.965,13.459-26.326,20.711c-3.617,3.137-6.654,7.5-8.541,11.91

          c-1.699,3.972-4.279,5.582-7.27,3.844c-4.216-2.451-
9.641-5.711-10.779-9.701c-1.008-3.535,2.527-8.971,5.218-12.738

          c1.68-2.352,5.36-3.684,8.406-4.581c19.875-
5.856,33.127-19.745,44.413-36.086c7.523-10.891,15.047-
21.782,22.42-32.774

          c2.904-4.328,6.319-6.044,11.818-
5.735c15.709,0.881,31.466,0.841,47.187,1.547c3.246,0.146,7.021
,1.174,9.53,3.116

          C647.988,527.369,668.798,553.342,686.37,583.229z"/>

     <path fill-rule="evenodd" clip-rule="evenodd"
d="M689.25,536.803c-17.811-8.838-30.9-20.341-43.713-32.189

          c-7.516-6.95-14.985-13.963-22.161-21.257c-1.632-
1.659-2.933-4.678-2.747-6.935c1.702-20.601-3.541-27.881-
23.928-32.465

          c-0.389-0.087-0.689-0.562-1.247-1.04c3.771-
6.021,10.402-7.028,15.061-
4.292c23.674,13.91,50.101,11.559,75.7,14.185


     c1.241,0.128,2.492,0.232,3.715,0.472c20.724,4.068,24.922,9
.953,18.717,29.896C703.116,500.95,696.034,518.24,689.25,536.80
3z"/>

     <path fill-rule="evenodd" clip-rule="evenodd"
d="M608.514,279.325c-0.435,6.746-0.054,11.764-1.222,16.392

          c-1.755,6.958-4.808,13.437-2.112,21.044c0.572,1.619-
2.701,5.248-4.936,7.03c-2.178,1.736-5.295,2.292-9.531,3.975

          c-2.988-10.719-6.104-20.572-8.375-30.616c-1.074-
4.753,1.157-8.524,6.454-
10.204C594.45,285.151,599.898,282.695,608.514,279.325z

          "/>

     <path fill-rule="evenodd" clip-rule="evenodd"
d="M704.912,620.806c-6.033,2.789-10.269,0.772-13.58-3.483

          c-4.636-5.961-4.049-18.012,0.873-23.811c3.973-
4.678,11.186-5.162,15.115-0.576c0.869,1.014,0.492,4.894-
0.337,5.285
```

```
          c-7.97,3.765-6.575,9.215-
3.58,15.445C704.42,615.777,704.43,618.372,704.912,620.806z"/>
</g>
</svg>
```

Anhang A 2. Ehrenwörtliche Erklärung

Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich meine Hausarbeit mit dem Titel

_____

_____

selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie nicht an anderer Stelle als Prüfungsarbeit vorgelegt habe.

_____
Ort

_____ _____
Datum Unterschrift