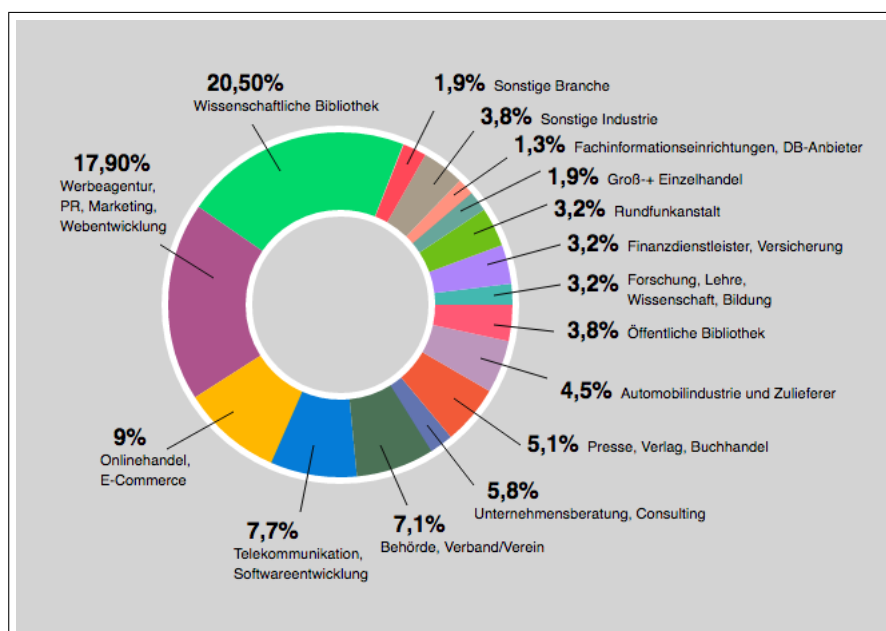


Erstellen einer Funktionalen Web Animation

Infografik „Informationsmanager auf dem Arbeitsmarkt“

Ein Tutorial für die Prüfungsleistung im
Modul: Medien Integration
Teilmodul: Oberflächen multimedialer Systeme



Erstellt von:
Frieda Josi (Matrikel-Nr. 1378361)

Prüferin:
Prof. Dr.-Ing. Monika Steinberg
14.05.17, Hannover

1 Aufgabenstellung

Entwicklung und Implementierung einer funktionalen Web Animation mit HTML5/CSS3 und/oder JavaScript mit API bzw. mit Werkzeugen eigener Wahl. Die Abgabe der Prüfungsleistung erfolgt auf weblab.wp.hs-hannover.de. Der Beitrag dazu befindet sich unter: <http://weblab.wp.hs-hannover.de/2017/05/17/informationsmanager-arbeitsmarkt-josi/>.

2 Einsatzzweck und Einordnung in Animationsart

Diese Infografik mit dem Inhalt „Informationsmanager auf dem Arbeitsmarkt“ dient der effektiveren Informationsvermittlung für, beispielsweise, der Webseite des Studienganges Informationsmanagement der Hochschule Hannover. Mit Hilfe der Animation wird das visuelle Erfassen der einzelnen Branchen, in denen Informationsmanager arbeiten, erleichtert. Die Daten, die für diese Infografik verwendet wurden, stammen aus der Absolventenbefragung 2016 von Prof. Dr. Jutta Bertram.¹

Diese Infografik ist ein Kreisdiagramm und besteht aus Micro-Interaktionen wie z.B. Click, Hover und Mouse out und verschiedenen Übergängen (Transitions).

3 Verwendete Bibliotheken, APIs, weitere Ressourcen

3.1 HTML5

HTML5-Seite, die die benötigten Container und Texte (Beschriftungen der Branchen und der Prozentsätze) für die Infografik, bereitstellt.

3.2 CSS3

In der CSS3 Datei sind die animierten Übergänge: `transition-delay`, `transform-origin`, `transition: opacity`, `transform: translate()`, `transition-duration`, `.button:hover` für die Infografik und für den „Animation“-Button angelegt. Ebenso die Formatierungen für HTML-Seite, Texte, Button und Infografik.

3.3 JavaScript:

In der JavaScript-Datei befinden sich die Funktionen für das Ausblenden der Infografik: `.addEventListener('click', function(e) ...)` und das Einblenden der Infografik: `setTimeout(function())`.

¹http://f3.hs-hannover.de/fileadmin/media/doc/f3/Studium/Bachelor/Bachelor_IK/bim/Informationsmanager_auf_dem_Arbeitsmarkt.pdf

4 Vorgehensweise

4.1 Verwendete Vorlagen

4.1.1 Animationsvorlage

Die Vorlage zu dieser funktionalen Animation wurde von Joshua Dorenkamp auf codepen.io zur Verfügung gestellt.² Für das Kreisdiagramm hat er acht Teile verwendet (siehe Abb. 4.1).

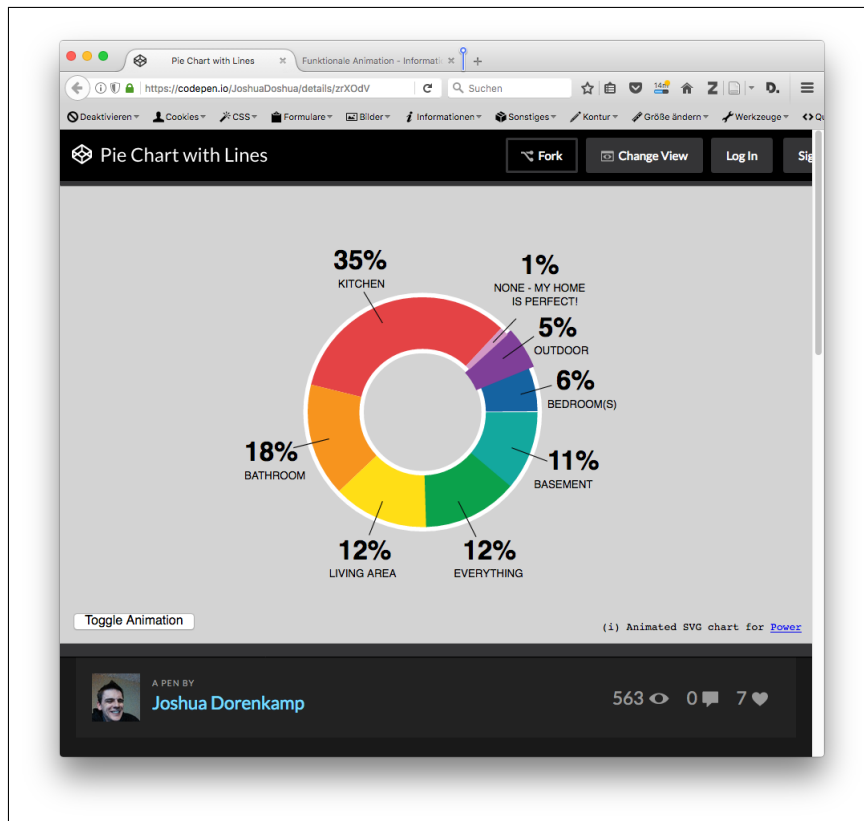


Abbildung 4.1: Original der funktionalen Animation von Joshua Dorenkamp

4.1.2 Buttonvorlage

Die Gestaltung des Buttons „Animation“ wurde mit einer Vorlage von w3schools.com erstellt.³

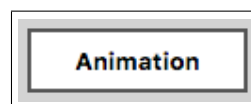


Abbildung 4.2: Animations-Button nach eigenen Anpassungen

4.2 Vorgenommene Anpassungen an der Grafik

Zu den acht vorhandenen Teilen des Kreisdiagrammes wurden noch weitere acht hinzugefügt.

²<https://codepen.io/JoshuaDoshua/details/zrXOdV>

³https://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_hover

Für jeden der 16 Teile in der CSS-Datei:

1. Je eine farbige Fläche und ihre Drehung:

```
.pie-1 {  
  stroke-dasharray: 0 628.32;  
  stroke-dashoffset: 388;  
  stroke: #01D86A;  
  -webkit-transition-delay: 25ms;  
  transition-delay: 25ms; }
```

2. Je eine Größe einer farbigen Fläche:

```
.animated .pie-1 {  
  stroke-dasharray: 134 628.32; }
```

3. Je eine Linie zu ihrer farbigen Fläche hin:

```
[class^="line-"].line-1 {  
  -webkit-transition-delay: 800ms;  
  transition-delay: 800ms; }
```

4. Das Erscheinen der farbigen Fläche: Wird im Abschnitt „4.4 Änderungen der Grafik im Detail“ unter 4.4.1/ Drehung aufgezeigt.

5. Das Erscheinen der Linie:

```
[class^="line-"].line-1 {  
  -webkit-transition-delay: 800ms;  
  transition-delay: 800ms; }
```

6. Das Erscheinen der Beschriftung:

```
[class^="text-"].text-1 {  
  -webkit-transition-delay: 800ms;  
  transition-delay: 800ms; }
```

7. Die Formatierung der Beschriftung:

```
.text-1bl {  
  font-size: 11px; }
```

Für jeden der 16 Teile in der HTML-Datei:

Je eine Prozentzahl (z.B. 20,50%) und eine Beschriftung (z.B. Wissenschaftliche Bibliothek) dazu.

```
<svg viewBox="0 0 650 500">  
  <circle class="outline" r="100" cx="210" cy="200" />  
  
  <circle class="pie-1" r="100" cx="210" cy="200" />  
  <line class="line-1" x1="165" y1="100" x2="144" y2="65" />  
  <text class="text-1 text-head" x="110" y="40">20,50%</text>  
  <text class="text-1 text-1bl" x="100" y="55">Wissenschaftliche  
  Bibliothek</text>
```

In der JavaScript-Datei wurden keine Änderungen vorgenommen.

4.3 Vorgenommene Anpassungen am Button

1. Formatierung des Buttons

```
.button {
    background-color: white;
    border: 2px solid #555555;
    color: black;
    padding: 7px 23px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 13px;
    font-weight: bold;
    margin: 0% 10%;
    -webkit-transition-duration: 0.4s; /* Safari */
    transition-duration: 0.4s;
    cursor: pointer; }
```

2. Formatierung des Buttons bei Mouse over

```
.button:hover {
    background-color: #555555;
    color: white; }
```

4.4 Änderungen an der Grafik im Detail

4.4.1 Für die Fläche eines einzelnen Teils aus dem Kreisdiagramm:

Drehung:

```
.pie-1 {
    stroke-dasharray: 0 628.32; /*Randlinie und Anzeige der Flaeche*/
    stroke-dashoffset: 388; /*Abstand des dasharray Musters*/
    stroke: #01D86A; /*Farbwert */
    -webkit-transition-delay: 25ms; /* Einblenden */
    transition-delay: 25ms; }
```

Größe:

```
.animated .pie-1 {
    stroke-dasharray: 134 628.32; }
```

Vergößerung bei Mouse over:

```
[ class ^=" pie " ]:hover
{
    stroke-width: 70;
    border-radius: 4px; }
```

Position:

Die Position wird bestimmt durch die Angabe der Drehung in der Kreislinie. Siehe oben unter „Drehung“ (stroke-dashoffset: 388; /*Abstand des dasharray Musters*/).

4.4.2 Für die Linie einer Zuordnung von Beschriftung und farbiger Fläche im Kreisdiagramm:**Länge, Drehung und Position in der HTML-Datei:**

```
<line class="line -1" x1="165" y1="100" x2="144" y2="65" />
```

4.4.3 Beschriftung und Prozentangabe im Kreisdiagramm:**Position in der HTML-Datei:**

```
<text class="text -1 text-head" x="110" y="40">20,50%</text >
<text class="text -1 text-bl1" x="100" y="55">Wissenschaftliche
Bibliothek </text >
```

4.5 JavaScript-Animation

Die JavaScript-Animationen wurden, aus dem Original von Joshua Dorenkamp, unverändert übernommen.

Animation, die beim Mausklick auf den Button „Animation“ ausgelöst wird:

```
var btn = document.getElementById('toggle '); /* Button aus index.html */
var wrap = document.getElementById('wrap '); /* Kontainer der Tortengrafik */
btn.addEventListener('click ', function(e) /* Ereignisobjekt fuer Button .*/
{
    wrap.className = ""; /* Ausblenden der Tortengrafik , Klick auf Button */
    setTimeout(function () {
        wrap.className = "animated";
    },3000);
});
setTimeout(function () /* Einblenden der Tortengrafik , beim Laden der Datei
und als zweites Ereignis nach dem Klick auf Button "Animation" */
{
    wrap.className = "animated";
},1000);
```

4.6 Bildschirmansicht der erstellten funktionalen Animation

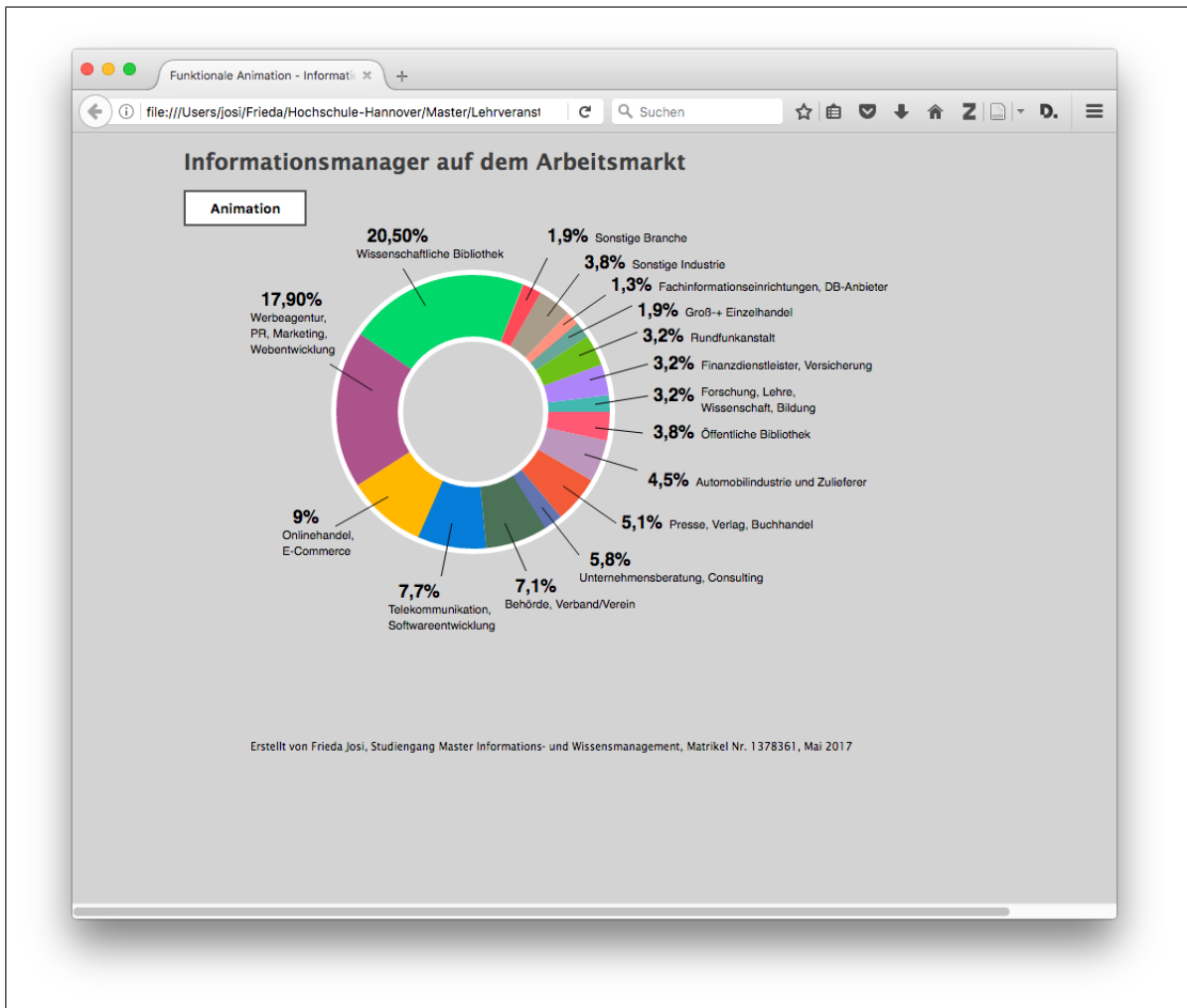


Abbildung 4.3: Funktionale Animation: Informationsmanager auf dem Arbeitsmarkt

Eine Onlineansicht kann unter: http://web_josi.wp.hs-hannover.de/infografik/ abgerufen werden.