

# PSMG

## INFORMATIONSVISUALISIERUNG

### UNIR



## Verzeichnis

1. Projektteilnehmer
2. Allgemeines
3. Starwars in a Nutshell
4. Explorative Möglichkeiten
5. Visuelle Kodierung
6. Softwarearchitektur
7. Aufgabenverteilung



Artan Ndreka - [artan.ndreka@stud.uni-regensburg.de](mailto:artan.ndreka@stud.uni-regensburg.de) (Matrikelnr. 1757069)

Silvia Ivanova - [silvia.ivanova@stud.uni-regensburg.de](mailto:silvia.ivanova@stud.uni-regensburg.de) (Matrikelnr. 1817043)

Robert Bosek - [robert.bosek@stud.uni-regensburg.de](mailto:robert.bosek@stud.uni-regensburg.de) (Matrikelnr. 1536516)

Stephan Jäger - [stephan.jaeger@stud.uni-regensburg.de](mailto:stephan.jaeger@stud.uni-regensburg.de) (Matrikelnr. 1865550)

## Allgemeines

Zuallererst stellte sich die Frage nach der Art des Datensatzes, der die Basis für eine Informationsvisualisierung für das PSMG bilden sollte.

Bei der Wahl des Datensatzes war wichtig, dass ...

... dieser Spaß macht

... dieser maximalen Freiraum für eine kreative Umsetzung der Visualisierung lässt

... dieser groß genug, aber dennoch überschaubar ist.

Schlussendlich entschied man sich gegen einen Datensatz mit realem Bezug, sowie ernster Thematik und wählte ein bekanntes SCI-FI Weltraumabenteuer, genauer StarWars, als Basis für das Abschlussprojekt.

Den Grundstein für diese Visualisierung legte die Internetseite „[www.swapi.co](http://www.swapi.co)“, die eine API als Zugang zu einer großen Menge an Daten des StarWars Universums bietet. Dieser Datensatz beinhaltet sowohl oberflächliche Informationen zu Planeten, Raumschiffen, Charakteren, deren Auftauchen in den Filmen, usw., als auch Detailwissen zu bspw. körperlichen Merkmalen einzelner Figuren.

(Die genaueren Elemente der Daten, sowie deren visuelle Kodierung werden im Laufe dieser Dokumentation noch genauer erläutert.)

Zur besseren Organisation untereinander bei der Zusammenarbeit im Team wurde Github, ein Onlinedienst zur Verwaltung von Software-Entwicklungsprojekten, verwendet. Github wurde in diesem Projekt eingesetzt, um den Code zentral verwalten zu können und eine gemeinsame Entwicklung im Team zu ermöglichen. Mit Hilfe dieser Plattform hielt man sich stets auf dem aktuellen Stand und trieb somit das Projekt Stück für Stück voran.

## Starwars in a Nutshell

### Die dunkle Bedrohung

Diese Trilogie startet damit, dass in der Galaktischen Republik ein Streit über die Besteuerung der Handelsrouten ausgebrochen ist. Der friedliche Planet Naboo mit seiner Königin Amidala wird von der geldgierigen Handelsföderation angegriffen. Im Mittelpunkt stehen Jedi Obi Wan Kenobi sowie der junge Anakin Skywalker, der sich als talentierter, potenzieller Jedi entpuppt.

### Angriff der Klonkrieger

Königin Amidala hat den Thron von Naboo verlassen, um Senatorin zu werden. Im Senat deckt sie einen Korruptionsdeal auf, und ihr Leben gerät in Gefahr. Anakin, inzwischen ein junger Jedi-Schüler, kommt zum ersten Mal in Kontakt mit der dunklen Seite der Macht.

### Die Rache der Sith

Die Kluft zwischen Kanzler Palpatine und dem Jedi-Rat wird immer größer. Dabei gerät der junge Jedi-Ritter Anakin Skywalker in einen Gewissenskonflikt. Palpatine entpuppt sich schließlich als dunkler Sith-Lord, der Anakin auf seine Seite zieht und zu Darth Vader werden lässt.

### Krieg der Sterne

In einer entfernten Galaxie wehren sich Rebellen gegen die Unterdrückung durch das machthungrige Imperium. An der Spitze der dunklen Streitkräfte steht die Figur Darth Vader mit ihrer schwarzen Atemmaske. Später wurde der Film in „Krieg der Sterne: Episode IV - Eine neue Hoffnung“ umbenannt.

### Das Imperium schlägt zurück

Darth Vaders gefährlichste Waffe, der Todesstern, ist zerstört, doch seine Truppen sind nicht geschlagen. Sie stöbern die Rebellen in ihren Verstecken auf und unterwerfen sie.

### Die Rückkehr der Jedi Ritter

Wesen aus allen Ecken der Galaxis treffen die Vorbereitungen für die entscheidende Schlacht gegen das Imperium - und gewinnen mit Hilfe der wiedererstarkten Jedi-Ritter.

### Das Erwachen der Macht

Das Imperium ist durch die Erste Ordnung abgelöst worden. Die ebenfalls diktatorische Organisation führt Krieg gegen den Widerstand. Beide Seiten suchen nach dem verschollenen Jedi-Ritter Luke Skywalker.

### Rogue one

Erneut geht es um den Kampf zwischen der Rebellion und dem zerstörerischen Imperium. Das Ende des Films knüpft nahtlos an das erste Werk von 1978 an.

## Exploration

Die explorativen Möglichkeiten der Visualisierung unterteilen sich in zwei große Teilbereiche. Zum einen die Hauptseite „Universe“, auf der der Besucher grundlegende Informationen bzw. Beziehungen erhält, und zum Anderen eine Unterseite „People“ mit detaillierteren Informationen der einzelnen Charaktere. Die Funktionen beider werden im Folgenden genauer erläutert.

### “Universe“

Auf dieser Seite erhält der Besucher grundlegende Informationen zu verschiedenen Elementen des Starwars-Universums. Die Seite besteht grundlegend aus 5 Teilsegmenten (Episoden, Planeten, Raumschiffe, Fahrzeuge, sowie Charakteren), die miteinander verlinkt sind. Wählt der Nutzer eines der gezeigten Elemente an, so werden ihm alle Elemente gezeigt, die mit dem ausgewählten in Verbindung stehen, indem nicht zugehörige Elemente ausgegraut werden.

Außerdem werden dem Nutzer durch Maus-„hovering“ Informationen zum jeweiligen Element im Tooltip angezeigt die bei Anwählen eines Elements im linken Screenbereich fixiert werden.

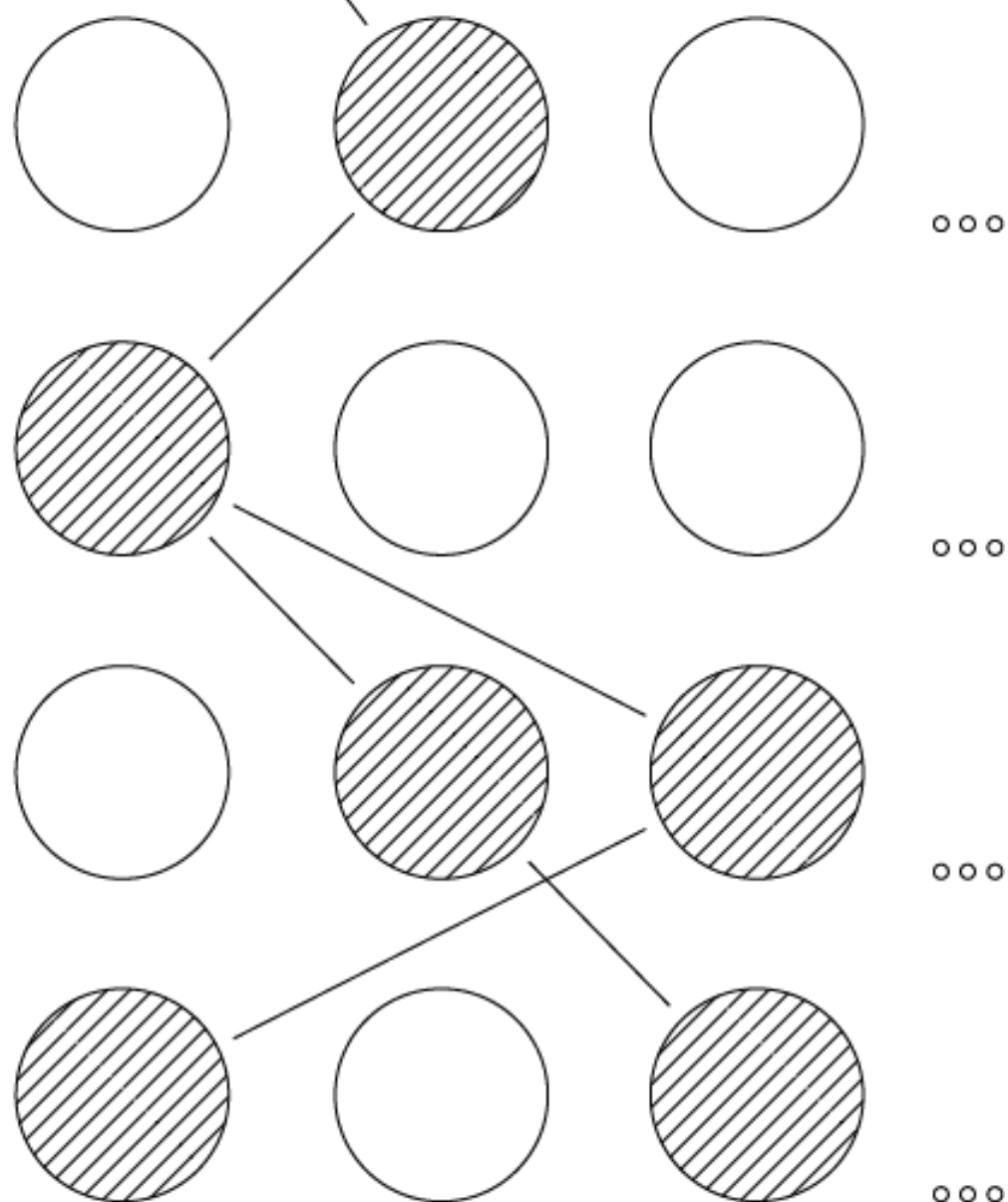
Zur Veranschaulichung der Handhabung ein Beispielszenario:

Ein eingefleischter Starwars-Fan besucht die Internetseite, um sich dort genauer über die einzelnen Planeten, Raumschiffe, Vehikel und Charaktere, sowie deren Zusammengehörigkeit im Starwars-Universum zu informieren.

Auf der Seite angekommen findet er im Kopf sieben Filmplakate, eines für jede Episode. Um herauszufinden, welche Elemente in welchem Film auftauchen, hat der Besucher nun die Möglichkeit den Inhalt der Seite über die Auswahl eines der im Header klickbaren Filmposter zu filtern und damit zu explorieren. Ebenso zeigt die Auswahl von z.B. „Anakin Skywalker“ alle anderen Elemente mit denen dieser Charakter in Verbindung steht.



Informationen ...



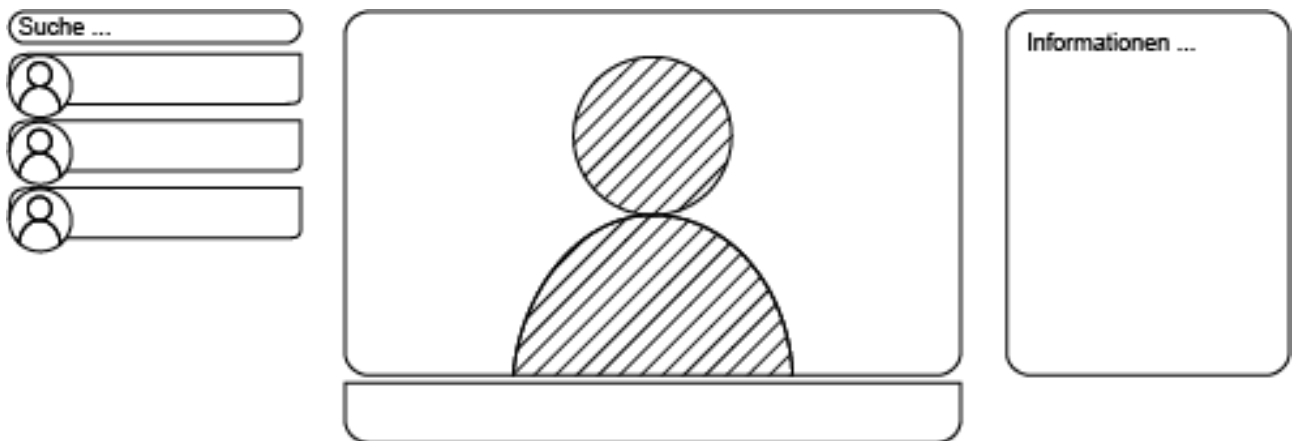
## People

Auf dieser Subseite hat man die Möglichkeit sich detaillierter über die in Starwars vorkommenden Charaktere bzw. deren Aussehen, körperliche Merkmale wie Haar-/Hautfarbe, sowie deren Fraktionszugehörigkeit und Geschlechterverteilung zu informieren.

Man erhält hier nicht nur Informationen darüber, welcher Charakter welche Merkmale trägt, sondern auch darüber, wie sich diese Merkmalsausprägungen innerhalb ihrer Gruppierung prozentual verteilen. Grundlegend besteht diese Seite aus einer Suchfunktion mit Darstellung der gesuchten Inhalte einerseits und andererseits aus einem Donutchart zur Visualisierung der Verteilungen der Merkmalsausprägungen.

### Teil 1 – suchfunktion

Den Start der „People“-Seite bildet die Suchfunktion. Mit Hilfe der Suchleiste im oberen linken Bereich hat der Nutzer die Möglichkeit, gezielt nach Charakternamen zu suchen. Die Liste filtert nach jedem eingegebenen, weiteren Buchstaben die möglichen Treffer. Hat er den gesuchten Charakter gefunden, so kann durch „hovern“ über den jeweiligen Namen ein großes Detailbild, sowie Name und Informationen der jeweiligen Figur angezeigt werden. Klickt er auf den Namen, so „locked“ er die angezeigten Informationen, die durch einen weiteren Klick auf das X im oberen rechten Bereich des Bildes wieder „unlocked“ und damit geschlossen werden.



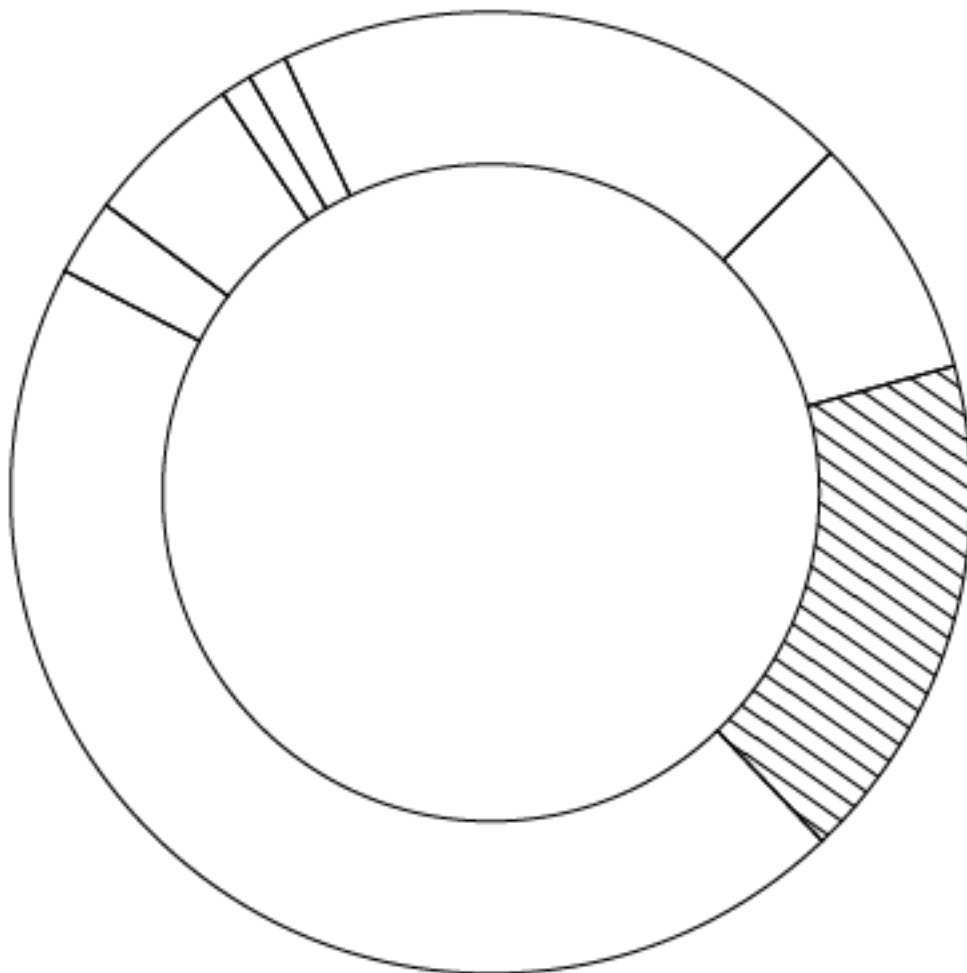


## Teil 2 - Donutchart

Die zweite Interaktionsmöglichkeit bietet die interaktive Donutchart, verknüpft mit oberhalb angeordneten Radiobuttons, die den Datensatz zur Anzeige im Chart filtern. Mit Hilfe dieses Elements hat der Besucher die Möglichkeit, sich quantitative Daten, sowie die prozentuale Verteilung zu Geschlechter, Fraktionszugehörigkeiten und körperlichen Merkmalen anzeigen zu lassen.

Durch „Hovern“ eines Kreissegments erscheint die Information dieses Segments unterhalb des Charts in Form eines Tooltips. Hier wird nicht nur das Merkmal an sich, sondern ebenfalls die Anzahl, wie oft dieses Merkmal im Datensatz vertreten ist, angezeigt.

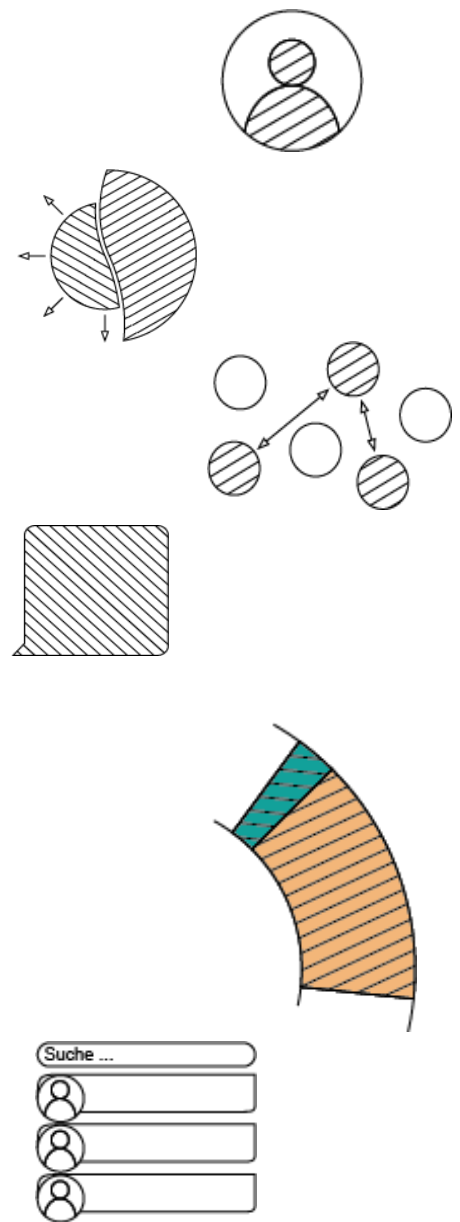
Da die Segmente die Daten lediglich „anonym“ visualisieren, sprich nicht aufzeigen, welcher Charakter nun genau diese eine bestimmte Merkmalsausprägung besitzt, sind die einzelnen Chartsegmente klickbar und wiederum mit der in Teil 1 beschriebenen Suchfunktion quer verlinkt. Ein Klick auf das jeweilige Segment filtert die unter der Suche angezeigte Liste an Charakteren. Auf diesem Weg erhält der Nutzer zu einem bestimmten Merkmal eine Liste derer, auf die dieses zutreffen und damit auch Fotomaterial um diese visuell identifizieren zu können.



*„Merkmalsausprägung: Anzahl“*

## Visuelle Codierung

Information:	Umsetzung:
<ul style="list-style-type: none"> <li>• Episoden</li> <li>• Planeten</li> <li>• Raumschiffe</li> <li>• Fahrzeuge</li> <li>• Charaktere</li> </ul>	<ul style="list-style-type: none"> <li>• Bilddateien (.png/.jpg)</li> </ul>
<ul style="list-style-type: none"> <li>• Planetengröße</li> </ul>	<ul style="list-style-type: none"> <li>• Planeten skalieren im Verhältnis zum 2. größten Planeten</li> </ul>
<ul style="list-style-type: none"> <li>• Gemeinsamkeiten</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgrauen der Items die nicht mit dem gewählten Element in Verbindung stehen</li> </ul>
<ul style="list-style-type: none"> <li>• Iteminformation</li> </ul>	<ul style="list-style-type: none"> <li>• Tooltip (Foto und/oder Textfeld)</li> <li>• Ausgelöst durch „Hover“-Effekte oder Klicks</li> </ul>
<ul style="list-style-type: none"> <li>• Haarfarbe</li> <li>• Augenfarbe</li> <li>• Hautfarbe</li> <li>• Geschlecht</li> <li>• Fraktionszugehörigkeit</li> </ul>	<ul style="list-style-type: none"> <li>• Jede Ausprägung als Teilsegment eines Donutcharts</li> <li>• Farbgebung der Segmente auf Merkmalsausprägung abgestimmt</li> <li>• Ein- und Ausblenden textueller Segmentinhalte durch „Hover“-Effekte</li> </ul>
<ul style="list-style-type: none"> <li>• Charaktere</li> </ul>	<ul style="list-style-type: none"> <li>• Durch Texteingabe filterbare Liste</li> </ul>



### Größencodierung für Planeten:

Da der größte Planet mit einem Durchmesser von 118k ca. 6,5x so groß ist wie der zweitgrößte, wurde die Entscheidung getroffen die Planeten relativ zum Zweitgrößten zu skalieren. Außerdem sind im Datensatz zerstörte Planeten (Durchmesser = 0) sowie Planete mit unbekanntem Durchmesser enthalten.

Deshalb wurde für die Größencodierung in Relation zum zweitgrößten Planeten folgendes festgelegt:

- 40% für zerstörte Planeten
- 50% für Planeten mit unbekannter Größe
- 100% für die beiden größten Planeten
- 50% + Größenrelation zum zweitgrößten Planeten für alle anderen Planeten

## Softwarearchitektur

“universe“

Das MVC-Pattern bildet die Grundlage der Softwarearchitektur der Subseite „Universe“.

Hierbei übernimmt die Klasse *UniverseAdministration.js* die Verbindung von *UniverseModel.js*, *UniverseController.js* und *UniverseView.js*. Außerdem sind dieser Klasse die Subklassen *film.js*, *planet.js*, *stашip.js*, *vehicle.js* und *character.js* untergeordnet.

Bei Seitenaufruf werden im Model alle Daten der Datenbank swapi.co mit der Bibliothek *d3* eingelesen. Mit Hilfe der genannten Subklassen werden aus den Daten JavaScript-Objekte erstellt, welche in weiteren Initialisierungsschritten miteinander verknüpft werden, um spätere Nutzerinteraktion effizient verarbeiten zu können. Als finalen Initialisierungsschritt werden die Daten durch die View-Klasse visualisiert. Hierbei wird für jedes Element des Datensatzes unter Zuhilfenahme der *d3*-Bibliothek eine Bilddatei gezeichnet, die dem Nutzer im Folgenden Interaktionsmöglichkeiten bietet. Die relative Anzahl an ausgewählten Elementen pro Kategorie wird außerdem durch ein Lichtschwert visualisiert, dessen Laserlänge mit dieser Anzahl skaliert. Außerdem werden dem Nutzer durch *hover*-Effekte zusätzlich zum dargestellten Bild weitere Informationen zum fokussierten Objekt angezeigt. Dies wurde mit Hilfe der *underscore*-Bibliothek und erstellten Templates umgesetzt. Zur Laufzeit wird das fokussierte Element und somit das zugrundeliegende Objekt, sowie die zu zeigenden Details ermittelt. Diese Details werden an eine mit *underscore* und dem *html*-Template erstellte *template*-Funktion gegeben und schließlich im Tooltip angezeigt.

Die Nutzerinteraktion wird über die Controller-Klasse verarbeitet. Klickt der Nutzer auf ein Element werden im Model zuerst sämtliche Objekte deaktiviert und im Anschluss nur dies Objekte aktiviert, die mit dem ausgewählten Element verbunden sind. Anschließend aktualisiert die View-Klasse anhand der im Model hinterlegten Daten die Visualisierung. Außerdem wird im linken Screenbereich ähnlich dem *hover*-Effekt eine Informationsanzeige mit den Details zum ausgewählten Element dargestellt.

“people“

Auf der Seite „People“ steht dem User eine „Search“-Applikation zur Verfügung, mit Hilfe derer er sich über Charaktere und deren äußere Merkmale genauer informieren kann.

Die „Search“-Applikation bietet die Möglichkeit, nach bestimmten Charaktern zu suchen und gezielt deren Informationen in einer interaktiven Visualisierung darzustellen. Wird ein Charakter ausgewählt, so erscheint ein Bild der Figur und zusätzlich, in Form einer InfoBox, die spezifischen Informationen über Namen, Geschlecht, Größe, Haut-, Haar- und Augenfarbe. Diese Informationen setzen sich aus lokal gespeicherten Image-Dateien und den Daten der Starwars API zusammen. Die App besteht insgesamt aus fünf JavaScript-Dateien, die die verschiedenen Komponenten implementieren.

### **PersonApp.js**

PersonApp dient als zentrale Komponente der Anwendung und steuert die übrigen Bestandteile. Das globale Objekt PersonApp, in dem das Modul gespeichert wurde, wurde als Namespace für die übrigen Module verwendet. Beim Starten der Anwendung werden die weiteren Module durch Aufruf der init-Methode initialisiert.

Die Kommunikation zwischen den übrigen Programmbestandteilen ist voneinander unabhängig. Diese geben die nötigen Informationen an das zentrale Modul weiter, bzw. werden von diesem sowohl informiert als auch gesteuert. Der Zugriff auf DOM-Elemente anderer Module wird hier als Referenz an die anderen Module weitergegeben.

### **PersonModel.js**

Dieses Model speichert den aktuellen Zustand der App. Hier wird die Liste der verfügbaren Personenobjekte gespeichert. Die JSON-Datei, auf die in diesem Modul zugegriffen wird, wird über eine synchronisierte HTTP-Anfrage an einen lokalen Server initialisiert und erneut verarbeitet. Um Probleme mit der Datenvisualisierung zu beheben, werden einige Elemente der JSON-Datei zur Laufzeit verändert.

Der Inhalt der ursprünglichen JSON-Datei bleibt dabei unverändert.

Auch die Funktionen zur Bereitstellung einer Liste aller vorhandenen Personen, sowie der Verarbeitung einer gefilterten Teilliste an Personen anhand übergebener Suchstrings und dem Zurückgeben aller Personen, deren Namen diesem Suchstring entsprächen, kommunizieren mit dem Model ebenfalls über ein öffentliches Interface.

### **PersonController.js**

Hier wurden alle relevanten Benutzerinteraktionen abgefangen und die damit verbundene Informationen weitergegeben. Hierzu zählen die Nutzereingaben im Suchfeld, die Auswahl eines Eintrags aus dem Menü sowie die Ereignisse bei „mouseover“, „mouseout“ und „click“. Zu jedem Ereignis wurden entsprechenden Listener zugeordnet.

In diesem Modul werden sowohl die Farben der jeweiligen Elemente geändert, sowie Art und Weise der Anzeige der Elemente der Galerie (Bilder und Datenfelder) aus dem Menü angepasst, je nachdem, ob das Objekt durch Klicken ausgewählt oder nur gehovert wurde. Jede Interaktion wird dadurch implementiert, dass ein entsprechendes HTML-Element ausgewählt wird. Anschließend wurden die entsprechenden Listener in Arrays gespeichert. Je nach Art der Interaktion werden die notwendigen HTML-Elemente hinzugefügt oder entfernt. Der Benutzer hat die Möglichkeit, die Informationen neben dem Menü anzeigen zu lassen, entweder durch Bewegung des Cursors auf das entsprechende Element (mouseover) oder durch Klick eines Menüeintrags. Durch Klicken wird eine statische Bilderansicht auf der rechten Seite des Menüs dargestellt welche durch Klick auf das

„X“ im oberen rechten Bereich des Bildes wieder entfernt werden kann.

### **PersonView.js**

Dieses Modul dient zur Initialisierung der Menüvorlagen und des Galeriebereichs. Abhängig von der Identifikationsnummer jeder Person aus dem Menü wird ein entsprechendes Foto angezeigt. Zu diesem Zweck wird für jede Instanz ein neues HTML-Element (div-Element) erstellt.

### **PersonListView.js**

Hier wird die View initialisiert. Der übergebenen Template-String wird für die weitere Verwendung vorbereitet und die Referenz auf den Listen-Container in einer Variable gespeichert. Die übergebenene Person wird dann zur Liste hinzugefügt. Dieses Objekt wird zusammen mit dem Template dazu verwendet, ein neues DOM-Element für die Liste zu erstellen.

### **people.html**

In dieser HTML-Datei wird der Inhalt der Seite strukturiert. Dabei wird das Template umgesetzt und die Pfade zu jedem Bild angelegt. Die Benennung der Bilder stimmt mit der ID der jeweiligen Person in der JSON-Datei überein. Die Namen der übrigen Bilddateien (die Symbole im Informationsfeld) entsprechen den Werten der JSON-Datei. Zusätzlich werden in dieser HTML-Datei die Radiobuttons initialisiert und mit Onclick-Listenern versehen. Diese Listener modifizieren den Inhalt des Donutcharts mit den jeweiligen, vorher strukturierten, übergebenen Daten.

“people-piechart“

Auf der letzten Seite „People-Piechart“ kann sich der User nochmals zu den Merkmalsausprägungen informieren. Im Gegensatz zur vorherigen Seite, geht es hier allerdings weniger um die Merkmalsausprägungen eines einzelnen Charakters an sich, sondern darum, wie sich diese Merkmale prozentual über den gesamten Datensatz verstreuen. Hier lässt sich bspw. Vergleichen, wie viele weibliche es im Verhältnis zu männlichen und geschlechtslosen Charakteren gibt.

### **donutApp.js**

Nach Laden des Datensatzes, dem Inhalt der .json-Datei, wird dieser mit entsprechenden Funktion vorsortiert. Dies ist deshalb nötig, da das Betätigen eines Radiobuttons einen Onclicklistener mit dem jeweiligen Datensatz aufruft und damit die hinterlegte Methode namens onChange() ausführt.

Diese Methode erhält den anzuzeigenden Datensatz, aktualisiert die einzelnen Segmente der Donutchart und weist diesen Außerdem die richtigen Farbwertarrays zu, welche wiederum über die Id des geklickten Buttons ermittelt werden.

Für jedes Teilsegment der Chart wird im SVG Piechart ein neues Pfadelement angelegt und mit den nötigen Werten versehen.

Zusätzlich zur Darstellung der prozentualen Anteile erfährt der User durch Hovern der einzelnen Segment auch noch die Merkmalsausprägung und die Auftrittshäufigkeit.

## Aufgabenverteilung

Sämtliche Planung sowie Entscheidungen zur graphischen und programmatischen Umsetzung wurden im Team, zumindest jedoch in Rücksprache mit allen Teammitgliedern getroffen.

Für die Universe Subseite war zu Beginn geplant, die aktiven Elemente mit einer Sunburst-Visualisierung darzustellen. Da diese Visualisierung Hierarchiestufen im Datensatz benötigt, die durch die querverlinkten Elemente nicht gegeben sind, werden die aktiven Elemente nun durch Barcharts pro Kategorie dargestellt (Laserschwerter).

Da die unterschiedlichen Charaktere und ihre genaueren Eigenschaften am interessantesten sind, worauf auch der Fokus der Informationsvisualisierung lag, sind diese genauer in Form von Donutchart und durchsuchbarer Personenliste dargestellt.

Es war geplant die Donutchart mit der Personenliste zu verbinden, wodurch durch Auswahl von Kreissegmenten der entsprechenden Kategorien die Personenliste gefiltert werden sollte, um auch die konkret hinter den Daten stehenden Elemente zu explorieren. Die Kombination von Personenliste und Donutchart konnte jedoch nicht umgesetzt werden, weshalb diese je eine eigene Subseite bilden.

### Aufgabenliste

Hilfestellung beim Programmieren in allen Teilbereichen	Robert Bosek
Html Grundstruktur <ul style="list-style-type: none"><li>Universe</li><li>People</li></ul>	Robert Bosek Silvia Ivanova
„Universe“-Subseite <ul style="list-style-type: none"><li>Vollständige Implementierung</li><li>CSS-Styling der Informationen in Info-Tabelle und Tooltip</li></ul>	Robert Bosek Artan Ndreka
Searching Applikation – (People)	Silvia Ivanova
Donutchart – (People)	Stephan Jäger
Bildbearbeitung/Illustration sämtlicher verwendeter Imagedateien, sowie Dokumentation	Stephan Jäger
Dokumentation - allgemein	Stephan Jäger, Robert Bosek
Dokumentation - Systemarchitektur	Robert Bosek Silvia Ivanova Stephan Jäger