



Universität Regensburg

Philosophische Fakultät III
Sprach- , Literatur- und Kulturwissenschaften
Institut für Information und Medien, Sprache und Kultur (I:IMSK)
Lehrstuhl für Medieninformatik

Projektseminar Mediengestaltung I: Informationsvisualisierung
Modul: MEI-M05.3
SS 2018
Leitung: Florin Schwappach

FBI-Statistiken visualisiert

Nina Hösl
Matrikelnummer: 18 27 558
5. Semester B.A. Medieninformatik / Informationswissenschaft
E-Mail: nina.hoesl@stud.uni-regensburg.de

Jonas Jelinski
Matrikelnummer: 18 27 577
5. Semester B.A. Medieninformatik / Informationswissenschaft
E-Mail: jonas.jelinski@stud.uni-regensburg.de

Christian Lisik
Matrikelnummer: 18 677 00
4. Semester B.A. Medieninformatik / Polnische Philologie/ Informationswissenschaft
E-Mail: christian.lisik@stud.uni-regensburg.de

Abgegeben am 30.07.2018



Inhalt

1	Visuelle Codierung der Daten.....	3
1.1	Art der Daten	3
1.2	Codierung der unterschiedlichen Verbrechenarten	3
1.3	Codierung der geographischen Verteilung der Verbrechen	3
1.4	Codierung der relativen Verteilung innerhalb eines Staates	4
1.5	Codierung der Häufigkeit von Verbrechen im Zeitverlauf	4
1.6	Codierung der Korrelation von Verbrechen	4
1.7	Verhältnis der Verbrechenskategorien	5
2	Dokumentation der explorativen Möglichkeiten.....	7
2.1	Map-Seite.....	7
2.2	Zeitverlauf.....	7
2.3	Korrelationen	7
2.4	Universum-Seite	8
3	Dokumentation der Softwarearchitektur	8
3.1	index.html und css	8
3.2	Charts und Pages	8
3.3	htmlElements.....	9
3.4	Store	9
3.5	Statemachine	10
3.6	Zuordnung der Pages zum DOM-ELEMENT.....	11
3.7	Scriptloader	12
4	Usability Tests	12
5	Browserkompatibilität	13
6	Aufgabenteilung.....	13
7	Literatur.....	14

1 Visuelle Codierung der Daten

1.1 Art der Daten

Es ist wichtig, sich vor Augen zu halten, welche Art von Daten unsere Website repräsentiert, um die Kodierung zu verstehen. Die Seite visualisiert Statistiken des Federal Bureau of Investigation (FBI) der Vereinigten Staaten von Amerika. Betrachtet werden unterschiedliche Verbrechenarten, wie Raub oder Mord, im Zeitraum von 2000 bis 2016 pro 100 000 Einwohner. Als Überkategorien der einzelnen Verbrechen gelten Gewalt- und Eigentumsdelikte.

1.2 Codierung der unterschiedlichen Verbrechenarten

Die FBI Statistiken beinhalten unterschiedliche Verbrechenarten. Jede Verbrechenart hat eine eindeutige Farbe, um sie von anderen Verbrechen zu unterscheiden. Diese Farben sind über alle Seiten hinweg konstant. Der Farbton jedes Verbrechens passt zum Farbton der jeweiligen Überkategorie.

1.3 Codierung der geographischen Verteilung der Verbrechen

Die Verteilung der Verbrechen über die Vereinigten Staaten ist heterogen. Sie gibt Aufschluss darüber, welche Art von Verbrechen wo am häufigsten geschehen. Für Nutzer, die sich für Verbrechen in den Vereinigten Staaten interessieren, ist die Verteilung von großer Bedeutung, da sich soziologische Schlüsse aus der Verteilung ziehen lassen.

Die Verteilung wurde mittels geometrischer Form und Farbe kodiert. Die geometrische Form ist eine Karte der Vereinigten Staaten, in der die einzelnen Staaten durch Linien voneinander abgegrenzt sind. Die Farbe zeigt die relative Häufigkeit der Verbrechen. Je roter der Staat ist, desto höher ist die Häufigkeit des Verbrechens.

Die Codierung erlaubt eine einfache Interpretation, da Nutzer verstehen, dass es sich bei der Karte um eine Repräsentation der USA handelt und eine höhere Verbrechensrate ein Anzeichen für Unsicherheit ist und allgemein als negativ angenommen wird, weshalb sich die Farbe Rot als Warnfarbe anbietet.

1.4 Codierung der relativen Verteilung innerhalb eines Staates

Innerhalb eines Staates treten die Verbrechen in unterschiedlicher Häufigkeit auf. Mord geschieht beispielsweise seltener als Diebstahl. Diese relative Verteilung wird über einen Sunburst dargestellt. Je häufiger ein Verbrechen auftritt, desto größer ist das repräsentierende Stück des Sunbursts. Der innere Ring des Sunbursts zeigt das Verhältnis zwischen Eigentums – und Gewaltverbrechen an, der äußere Ring die einzelnen Verbrechen. Durch diese Aufteilung ist schnell die Zugehörigkeit der Verbrechen zu ihren Klassen ersichtlich und ihr verhältnismäßiger Anteil.

Gleichzeitig steht eine Baumstruktur zu Verfügung, welche diese Zugehörigkeit noch einmal beschreibt. Für Nutzer, welche die genauen Zahlen wissen möchten, werden diese an den Blättern des Baumes angezeigt. Durch die doppelte Darstellung wird sichergestellt, dass der Nutzer die Struktur der Verbrechenskategorien versteht.

1.5 Codierung der Häufigkeit von Verbrechen im Zeitverlauf

Die Häufigkeit von Verbrechen nimmt über die Jahre hinweg betrachtet zu oder ab.

Für den Betrachter ist dies von großem Interesse, da er sich ein Bild über die Entwicklung der Verbrechen machen möchte.

Die Seite stellt ein Liniendiagramm dar, um den Zeitverlauf zu visualisieren, da dies ein Standard in verschiedenen wissenschaftlichen Disziplinen ist.

1.6 Codierung der Korrelation von Verbrechen

Verbrechen korrelieren unterschiedlich miteinander, um dies darzustellen verlinkt die Seite die einzelnen Verbrechen miteinander. Anhand der Farbe der Verbindungslinie zeigt sich wie hoch die Korrelation ist und ob sie sich im positiven oder im negativen Bereich befindet. Da jedes Verbrechen mit jedem korreliert, entsteht ein Polygon.

Diese Art der Kodierung erlaubt es, alle Korrelationen gleichzeitig zu betrachten.

Falls der Nutzer nur die Korrelationen einer Verbrechensart auswählen möchte, färben sich die irrelevanten Verbindungslinien in eine neutrale Farbe.

Falls den Nutzer die genaue Höhe der Korrelation interessiert, kann er sich auch die genaue Zahl anzeigen lassen.

Normalerweise werden Korrelationen graphisch dargestellt, indem die relevanten Parameter als Linien in einem Zeitdiagramm nebeneinandergelegt werden. Dies hat fol-

gende Nachteile: Erstens kann ohne statistische Tests mit rein graphischer Darstellung tatsächlich nichts über die Korrelation ausgesagt werden. Zweitens verliert bei sieben verschiedenen Verbrechen der Nutzer schnell die Übersicht. Drittens ist die genaue Höhe der Korrelation nicht ablesbar.

Unsere Darstellungsart gleicht diese Nachteile aus, da das Farbschema leicht zu verstehen ist und Menschen aufgrund der Wahrnehmungsgesetze Dinge einander zuordnen, die durch eine Linie miteinander verbunden sind.

1.7 Verhältnis der Verbrechenkategorien

Jede Verbrechenart wird einer von zwei Überkategorien zugeordnet. Entweder ist ein Verbrechen ein Eigentumsverbrechen oder es ist ein Gewaltverbrechen. In jedem Staat stehen diese Verbrechen in einem bestimmten Verhältnis zueinander. In den meisten Staaten gibt es mehr Eigentums-, als Gewaltverbrechen. Um zu sehen, in welchem Staat es wahrscheinlicher ist, Opfer eines Gewaltverbrechens als eines Eigentumsverbrechens im Vergleich zu den anderen Staaten zu werden, haben wir die Normalisierte Relation der Kategorierelationen graphisch dargestellt. Die Normalisierte Relation der Kategorierelationen (NR) berechnet für jeden Staat das Verhältnis (Q) von Gewaltverbrechen (G) zu Eigentumsverbrechen (E) und transformiert den Wert zwischen 0 und 1. 0 steht für die niedrigste gemessene Relation (minQ) in allen Staaten und 1 für die höchste (maxQ).

$$Q = \frac{G}{E}; \quad \text{delta} = \text{maxQ} - \text{minQ}; \quad \text{NR} = \frac{Q - \text{minQ}}{\text{delta}};$$

Diese komplexe Formel darzustellen ist mit herkömmlichen Graphen nicht möglich, daher zeigt die Seite die Verteilung mittels zweier Sonnensysteme an. Jeder Staat wird durch einen Planeten repräsentiert. Jede Kategorie durch eine Sonne im Zentrum des jeweiligen Sonnensystems. Je roter ein Planet ist, desto „gewalttätiger“ ist der Staat. Zusätzlich erhöht die Anzahl an Eigentumsdelikten die Anziehungskraft. Planeten, die nahe an einer Sonne liegen, haben mehr Eigentumsdelikte, als die Planeten, die im gleichen Sonnensystem weiter weg von der gleichen Sonne liegen. Ein Planet landet im Sonnensystem der Gewaltdelikte, wenn sein $Q > 0.5$ ist, anderenfalls kreist er um die Eigentumssonne. Je größer Q ist desto langsamer bewegt er sich.

Die Codierung wurde auf folgender Grundlage erstellt: Erstens ist ein Farbschema eine extrem intuitive Möglichkeit die sukzessive Änderung von Q zu vermitteln. Zweitens kann der Nutzer durch die zwei Sonnensysteme mittels eines einzigen Blickes die 0.5 Marke ausmachen, sobald er das Konzept verstanden hat. Drittens ist die erhöhte Anziehungskraft ein einfacher Weg, die Planeten innerhalb eines Sonnensystems zu vergleichen, ohne dass dadurch Rückschlüsse auf die der anderen Sonnensysteme geschlossen werden müssen.

2 Dokumentation der explorativen Möglichkeiten

„Bei der explorativen Analyse findet häufig eine interaktive, ungerichtete Suche nach Informationen und Strukturen statt, ohne dass Benutzer im Detail erklären könnten, was überhaupt gesucht wird.“ (Preim & Dachsel, 2010 , S.442).

Die explorativen Möglichkeiten wurden an dem „Mantra visueller Informationssuche“ (Shneiderman , 1996) ausgerichtet:

1. Gewinnen eines Überblicks über den gesamten Informationsraum
2. Zoomen zum Betrachten einer kleineren Untermenge von Daten
3. Herausfiltern von uninteressanten Datenobjekten
4. Auswahl eines Datenobjektes oder einer Gruppe von Daten, um Details zu erhalten

2.1 Map-Seite

Beim Öffnen der Seite bekommt der Nutzer sofort einen Überblick über den gesamten Informationsraum, indem er einen Überblick über die Karte der USA erhält. Er kann Informationen filtern, indem er die Timeline zur Auswahl des Jahres oder das Dropdown-Menü zur Auswahl des Verbrechens erhält. Durch das Klicken auf einen Staat erhält der Nutzer Details zu dem ausgewählten Staat, indem ein Pop-Up Menü mit Details angezeigt wird.

2.2 Zeitverlauf

Der Zeitverlauf gibt einen schnellen Überblick über die Menge aller Verbrechen über mehrere Jahre hinweg. Der Nutzer kann mittels Scrollen zoomen, um eine Untermenge der Daten genauer betrachten zu können. Mittels der zwei Menüs kann der Nutzer zwischen den einzelnen Staaten wechseln und uninteressante Verbrechen herausfiltern. Zudem stehen ihm zwei Diagramme parallel zur Verfügung, um zwei Staaten direkt vergleichen zu können.

2.3 Korrelationen

Auf der Seite zum Anzeigen von Korrelationen bekommt der Nutzer eine schnelle Übersicht über die Korrelationen zwischen den einzelnen Verbrechen. Er kann näher an Objekte heranzoomen und den Graphen wenden und drehen, um eine Untermenge

von Daten genauer zu betrachten. Durch das Hovern über die Linie zwischen zwei Kreisen, kann der Nutzer sich Details zu der Korrelation anzeigen lassen. Durch das Hovern über eine Kategorie, kann der Nutzer das gewählte Verbrechen besonders hervorheben.

2.4 Universum-Seite

Auf der Universum-Seite kann der Nutzer einen Überblick über die Verteilung des Verhältnisses von Gewaltstraftaten zu Eigentumsstraftaten gewinnen. Über den Zeitstrahl kann der Nutzer das Jahr auswählen, für das er sich interessiert. Durch Hovern über einen Planeten wird der Name des Staates angezeigt, um Details der Verteilung zu erhalten. Planeten können durch zoomen vergrößert werden.

3 Dokumentation der Softwarearchitektur

Die Softwarearchitektur ist eine Mischung aus dem klassischen MVC-Pattern und dem Pattern, welches Redux verwendet¹.

3.1 index.html und css

Die `index.html` Datei gibt die Grundstruktur der Website durch HTML-Tags vor. Die `css`-Datei stellt die `css`-Klassen, um das Layout genauer zu definieren.

3.2 Charts und Pages

Die Komponenten der einzelnen Seiten werden in Javascript geschrieben. Die Hauptmodule sind Charts und Pages. Charts sind Klassen², deren Instanzen Daten visualisieren, zum Beispiel in der Form eines Sunbursts. Jede dieser Chart-Klassen erbt von der Klasse `MagicCircle`. Pages sind Klassen, deren Instanzen interaktive Seiten darstellen.

Dazu bedienen sie sich der Chartklassen. In den Pages-Klassen werden Instanzen der einzelnen Charts erzeugt, welche dargestellt werden sollen. Dort wird auch die Kommunikation zwischen diesen einzelnen Charts implementiert.

¹ Redux (2018). Three Principles. Einsehbar unter: <https://redux.js.org/introduction/three-principles>, zuletzt abgerufen am 26.06.2018

² Es handelt sich dabei nicht, um Klassen im OOP-Sinn, sondern die der ECMAScript 2015 - Syntax.

3.3 `htmlElements`

Die einzelnen Charts greifen auf die Objekte des Namespaces `htmlElements` zu. Darin befinden sich für jeden einzelnen Chart hinterlegt relevante Variablen, wie die Breite und Höhe oder der Containertyp, z.B. `SVG`, in dem der Chart dargestellt werden soll.

3.4 `Store`

Die Daten, welche die Charts brauchen, um die Informationen darzustellen, befinden sich im Ordner `Store`. Dazu zählen unter anderem die csv-Datei `crimeCorrelation.csv`, welche die Korrelationen zwischen den einzelnen Verbrechen im Tabellenformat enthält oder die JSON-Datei `crimes.json`, welche alle relevanten Daten der FBI-Kriminalstatistik im JSON-Format darstellt. Die Klassen greifen direkt auf diese Dateien zu. Ebenfalls enthält der `Store` die Config-Datei `config.js`, welche Werte, die klassenübergreifend relevant, aber keine FBI-Statistiken sind, innerhalb von Objekten speichert. Zum Beispiel sind die Farben, die für jedes Verbrechen hinterlegt sind, dort gespeichert.

Im `Store` befindet sich ebenfalls die Datei `commonfunctions.js`. Darin befinden sich Funktionen, welche klassenübergreifend relevant sind. Zum Beispiel greift die Funktion `getCrimeColor(crime)` auf die config-Datei zu und gibt die Farbe zurück, welche für ein bestimmtes Verbrechen hinterlegt ist. `commonfunctions.js` soll verhindern, dass Programmierer Code doppelt schreiben und erhöht die Lesbarkeit des Codes.

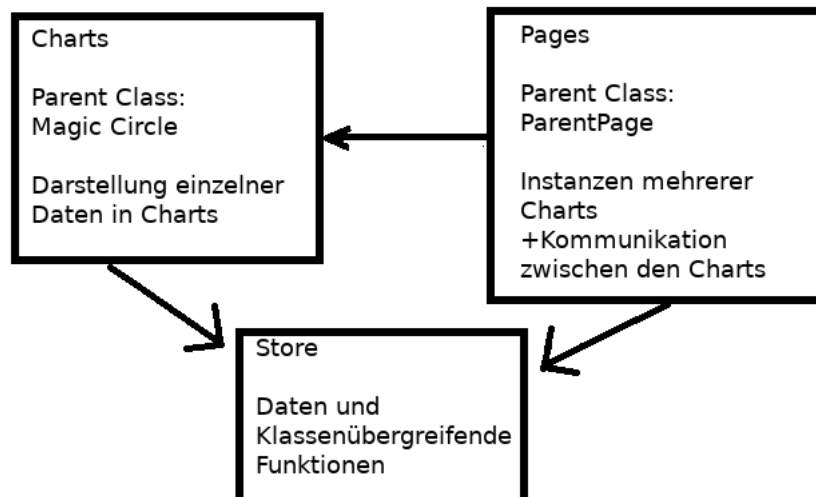


Abbildung 1 Pages enthalten die Instanzen der Charts. Beide greifen auf den Store zu.

3.5 Statemachine

Die einzelnen Seiten werden in einer StateMachine verwaltet. In der Klasse StateMachine werden die Instanzen der Seiten erstellt. In jedem Zustand wird eine andere Seite dargestellt. Der Zustand wird mittels der Methode `switchState(state)` gewechselt.

Die Instanz der StateMachine befindet sich in der Datei `components.js`. In der Datei `action.js` befinden sich die Funktionen, welche `switchState(state)` mit unterschiedlichen Zuständen als Parameter aufrufen und somit die StateMachine in einen neuen Zustand versetzen. Diese Actions werden in der Datei `listener.js` aufgerufen. In dieser Datei werden die DOM-Elemente mit Click-Listenern ausgestattet. Sobald ein DOM-Element geklickt wird, wird die passende Seite dargestellt, indem die passende Action aufgerufen wird und die StateMachine den Zustand wechselt.

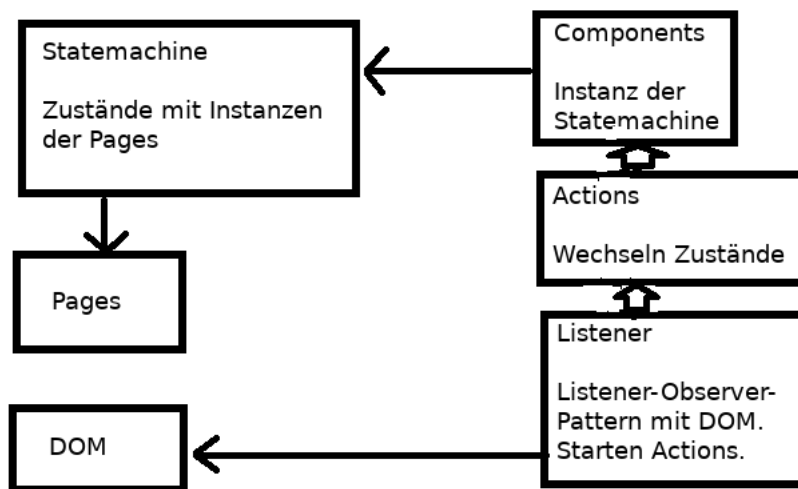


Abbildung 2 Der Aufruf der Seiten erfolgt nach dem Observer-Listener Pattern.

3.6 Zuordnung der Pages zum DOM-Element

Damit jede Page und jeder Chart weiß, in welchem Element des DOMs sie dargestellt werden soll, erhält jede Page im Konstruktor eine `id`. Diese `id` gibt sie an ihre Charts weiter. Die `id` ist die `css-id` des DOM-Elements. Mittels der `commonfunctions`-Methode `getPageById(id)` wird das DOM-Element herausgesucht und geladen.

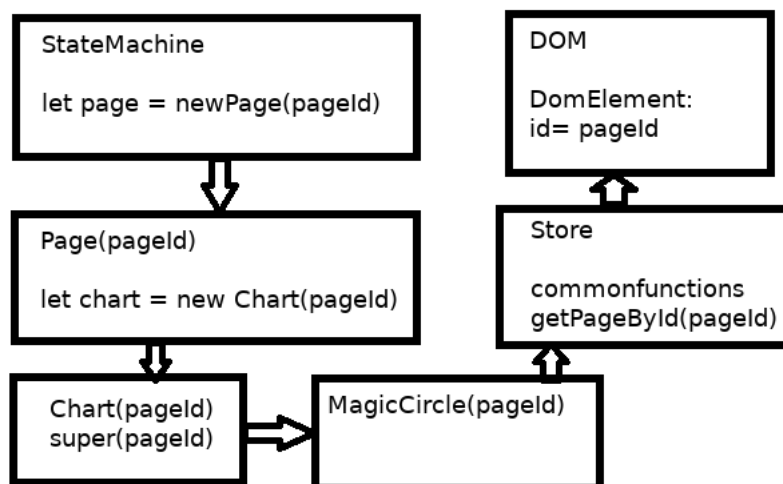


Abbildung 3 Die Zuordnung der Elemente zum DOM-Container erfolgt mittels einer `id`

3.7 Scriptloader

Das Scriptloader-Modul lädt alle Scripts und Bibliotheken, indem es auf die in der `scripts.js` hinterlegten Pfade zurückgreift. In der Methode `loadObjectsAfterWaitingForScripts()` ist ein Timer hinterlegt. Ist er abgelaufen, werden die JSON-Dateien und CSV-Dateien eingelesen und anschließend wird die Startseite erstellt. Dies ist notwendig, um sicherzustellen, dass alle Skripte vollständig geladen wurden, bevor mit ihnen gearbeitet wird. Der Scriptloader wird in der `index.html` aufgerufen, um die Anwendung zu initialisieren.

Charts	Enthält die Charts. Diese stellen Daten graphisch dar.
Pages	Enthält die Pages. Diese sind für die Instanziierung der Charts und die Kommunikation zwischen den Charts verantwortlich,
htmlelements	htmlelements.js. Enthält Layout-Daten für die Charts.
StateMachine	Enthält StateMachine, Components, Listener, Actions
Scripts	Enthält ScriptLoader und Pfade zu den Scripts und Bibliotheken.
Store	Enthält Daten, Config - und css Datei
Libs	Enthält Bibliotheken

Abbildung 4 Die wichtigsten Komponenten im Überblick.

4 Usability Tests

Vor der Abgabe wurde die Website drei Usability Tests mit Testpersonen im Alter zwischen 13 und 36 unterzogen. Die Seite wurde anschließend noch einmal überarbeitet und die aufgefallenen Mängel beseitigt.

5 Browserkompatibilität

Die Website wurde auf unterschiedlichen Computern mit unterschiedlichen Browsern getestet. Eine fehlerfreie Darstellung kann nur unter Mozilla Firefox 61.01 (64x) garantiert werden. Chrome und andere Browser verursachen Fehldarstellungen. Chromes Sicherheitseinstellungen sind zu restriktiv, um die benötigten Daten aus der Festplatte zu lesen. Weiterhin besitzt Chrome einen Bug ³, der dazu führen kann, dass das property `transform` nicht richtig gesetzt werden kann.

6 Aufgabenteilung

Komponente	Autor
.html	Nina Hösl
.css	Nina Hösl
Usability Tests	Nina Hösl
Statistik	Nina Hösl
bubbleMenu.js	Jonas Jelinski
closeButton.js	Jonas Jelinski
colorLegend.js	Jonas Jelinski
crimeCorrelation.js	Jonas Jelinski
dropDownMenu.js	Jonas Jelinski
infoBox.js	Jonas Jelinski
infoText.js	Jonas Jelinski
lineChart.js	Jonas Jelinski
magicCircle.js	Jonas Jelinski
map.js	Christian Lisik
playButton.js	Jonas Jelinski, Christian Lisik
sunburst.js	Jonas Jelinski
timeLine.js	Christian Lisik
tree.js	Christian Lisik

³Issue 701075: <https://bugs.chromium.org/p/chromium/issues/detail?id=701075>

universe.js	Jonas Jelinski, Nina Hösl
htmlElements.js	Jonas Jelinski, Christian Lisik
crimeCorrelationPage.js	Jonas Jelinski, Nina Hösl
dataRegulationPage.js	Jonas Jelinski
impressumPage.js	Jonas Jelinski
infoPage.js	Jonas Jelinski
lineChartPage.js	Jonas Jelinski, Nina Hösl
mapPage.js	Jonas Jelinski
parentPage.js	Jonas Jelinski
popUpPage.js	Jonas Jelinski
textPage.js	Jonas Jelinski
universePage.js	Jonas Jelinski, Nina Hösl
scriptLoader.js	Jonas Jelinski
scripts.js	Jonas Jelinski
actions.js	Jonas Jelinski
listener.js	Jonas Jelinski, Nina Hösl
commonfunctions.js	Jonas Jelinski
config.js	Nina Hösl, Jonas Jelinski, Christian Lisik
crimeCorrelation.csv	Nina Hösl
crimes.json	Christian Lisik
infoTexts.js	Nina Hösl

7 Literatur

Preim, Bernhard & Dachsel, Reimund (2010): Interaktive Systeme. Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer. Berlin Heidelberg

Shneiderman, Ben (1996): The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, S. 336-343, Washington. IEEE Computer Society Press