Erstellen von UI-Elementen mittels ReactJS

vorgelegt von

**Achim Strohm**Matrikelnr.: 24967

Betreuender Professor:

Prof. Dr.- Ing. Thorsten Weiss

Inhalt

[Abkürzungsverzeichnis 2](#_Toc485457757)

[Abbildungsverzeichnis 3](#_Toc485457758)

[1. Einleitung 4](#_Toc485457760)

[2. Begriffserklärungen 4](#_Toc485457761)

[2.1 UI 4](#_Toc485457762)

[2.2 CSS 4](#_Toc485457763)

[2.3 Bootstrap 4](#_Toc485457764)

[2.4 JavaScript 4](#_Toc485457765)

[2.5 IDE 4](#_Toc485457766)

[2.6 npm 5](#_Toc485457767)

[2.7 node.js 5](#_Toc485457768)

[2.8 ReactJS 5](#_Toc485457769)

[2.9 JSX 5](#_Toc485457770)

[2.10 Webpack 5](#_Toc485457771)

[2.10.1 Konkretes Ablaufbeispiel am verwendeten react-seed 6](#_Toc485457772)

[2.11 Git/GitHub 6](#_Toc485457773)

[3. Grundlagen 7](#_Toc485457774)

[3.1 rect-seed als Grundgerüst 7](#_Toc485457775)

[3.2 Einrichten der Projektumgebung 8](#_Toc485457776)

[3.2.1 Übersicht Installation git: 8](#_Toc485457777)

[3.2.2 Installation Node.js LTS 8](#_Toc485457778)

[3.2.3 Installation der Abhängigkeiten 9](#_Toc485457779)

[3.2.4 Zugriff eines Smartphones über das Homenetz 9](#_Toc485457781)

[3.2.5 Starten des Projekts 9](#_Toc485457782)

[Nun wird in der Konsole in das Projektv 9](#_Toc485457783)

[Im Browser auf dem PC oder auf dem Smartphone kann mit 9](#_Toc485457784)

[3.2.6 Erlernen der Grundlagen 10](#_Toc485457785)

[Zum Erlernen der Grundlagen kann das von Facebook bereitgestellte Tutorial und Dokumentation verwendet werden. 10](#_Toc485457786)

[3.2.7 React Developer Tools 10](#_Toc485457787)

[4. Umsetzung 11](#_Toc485457788)

[4.1 Ziel des Informatikprojekts 11](#_Toc485457789)

[4.2 Schrittweise Umsetzung von einem normalen Button zu einem fertigen Revolver Button 12](#_Toc485457790)

[5. Nutzertests 28](#_Toc485457791)

[6. Anpassungen des RevolverButtons nach den Nutzertests 30](#_Toc485457792)

[Literaturverzeichnis 32](#_Toc485457793)

[Selbständigkeitserklärung 33](#_Toc485457794)

Abkürzungsverzeichnis  
UI User Interface  
HTML Hyper Text Markup Language  
CSS Cascading Style Sheets [siehe 3.21]  
NPM Node Package Manager  
IDE Integrated Developement Environment [siehe 3.22]

Abbildungsverzeichnis

[Abbildung 1: Webpack 7](#_Toc485457866)

[Abbildung 2: Button zum Ändern des Styles 12](#_Toc485457867)

[Abbildung 3: Ausgeben des Buttons auf der StepOne.js Page (src/js/pages/StepOne.js) 13](#_Toc485457868)

[Abbildung 4: src/js/components/step\_1/tutorial/RevolverButton/Button.js 14](#_Toc485457869)

[Abbildung 5: Ansicht Schritt 1 im Browser 14](#_Toc485457870)

[Abbildung 6: Ausgeben des Buttons auf der StepTwo.js Page (src/js/pages/StepTwo.js) 15](#_Toc485457871)

[Abbildung 7: Schritt 2: Styling des Buttons 16](#_Toc485457872)

[Abbildung 8: Ansicht Schritt 2 im Browser 16](#_Toc485457873)

[Abbildung 9: Ausgeben des Buttons auf der StepThree.js Page (src/js/pages/StepThree.js) 17](#_Toc485457874)

[Abbildung 10: src/js/components/step\_3/RevolverButton/Button.js 19](#_Toc485457875)

[Abbildung 11: src/js/components/step\_3/tutorial/RevolverButton/RevolverItem.js 20](#_Toc485457876)

[Abbildung 12: Ansicht Schritt 3 im Browser 20](#_Toc485457877)

[Abbildung 13: Ausgeben des Buttons auf der StepFour.js Page (src/js/pages/StepFour.js) 21](#_Toc485457878)

[Abbildung 14: src/js/components/step\_4/tutorial/RevolverButton/Button.js 22](#_Toc485457879)

[Abbildung 15 : src/js/components/step\_4/tutorial/RevolverButton/RevolverItem.js 23](#_Toc485457880)

[Abbildung 16: Ansicht Schritt 4 im Browser 24](#_Toc485457881)

[Abbildung 17: Ausgeben des Buttons auf der StepFive.js Page (src/js/pages/StepFive.js) 24](#_Toc485457882)

[Abbildung 18: src/js/components/tutorial/step\_5/RevolverButton/Button.js 25](#_Toc485457883)

[Abbildung 19: src/js/components/step\_5/RevolverButton/RevolverItem.js 26](#_Toc485457884)

[Abbildung 20: Ausgabe Schritt 5 im Browser 26](#_Toc485457885)

[Abbildung 21: Ausgeben des Buttons auf der StepSix.js Page (src/js/pages/StepSix.js) 27](#_Toc485457886)

[Abbildung 22: src/js/components/step\_6/RevolverButton/Button.js 27](#_Toc485457887)

[Abbildung 23: src/js/components/step\_6/RevolverButton/RevolverItem.js 28](#_Toc485457888)

[Abbildung 24: Ausgabe Schritt 6 im Browser 28](#_Toc485457889)

[Abbildung 25: src/js/components/step\_6/RevolverButton/RevolverItem.js 31](#_Toc485457890)

[Abbildung 26: Ausgabe Schritt 7 im Browser 31](#_Toc485457891)

[Abbildung 27: Ausgabe im Browser 32](#_Toc485457892)

Alle Abbildungen die in dieser Dokumentation verwendet wurden, wurden von mir selbst erstellt.

1. Einleitung  
Ziel dieses Tutorials ist es, die Grundlagen bei der Erstellung von UI-Elementen mittels ReactJS leicht verständlich zu vermitteln.

# 2. Begriffserklärungen

2.1 UI  
Ein UI ist die Benutzeroberfläche. Sie ist die Schnittstelle zwischen Benutzer und Computer. Der Nutzer kann in dem UI Daten und Befehle eingeben, die anschließend vom Computer ausgeführt werden. Sie soll außerdem gewährleisten, dass Computerunerfahrene Nutzer durch einfache Gestaltung der UI trotzdem einen Computer bedienen können. [1]

2.2 CSS  
Mittels CSS können Gestaltungsanweisungen erstellt werden, die vor allem für das Styling von HTML-Elementen zum Einsatz kommt. Mithilfe von CSS können die Gestaltungsanweisungen von dem eigentlichen Inhalt, der über HTML angelegt wird getrennt werden. [2]

2.3 Bootstrap  
Bootstrap ist ein von Twitter entwickeltes frei zur Verfügung gestelltes CSS-Framework. Es basiert auf HTML5 und CSS3 und enthält Gestaltungsvorlagen für Typografie, Formular, Buttons, Grid-Systeme uvm. [3]

2.4 JavaScript

JavaScript ist eine Skriptsprache, die den Funktionsumfang von HTML und CSS erweitert. Mit ihr kann dynamisch HTML generiert werden und es können Benutzerinteraktionen wie z.B. die Validierung von Formularen vorgenommen werden. [4]

2.5 IDE

Eine IDE besteht meist aus einem Texteditor, Compiler bzw. Interpreter, Linker und Debugger. Mit einer IDE wird es möglich Software mit nur einem Anwendungsprogramm zu entwickeln. [5]

2.6 npm  
npm ist ein Paketmanager für node.js. Damit kann auf ein Repository mit über 350.000 Open Source Paketen (Stand 13. Januar 2017) zugegriffen werden. Über die Konsole können die Softwarepakete mit folgendem Befehl heruntergeladen werden:

$npm install %Modulname

Außerdem können Abhängigkeiten automatisch aufgelöst werden. Wenn ReactJS Programme auf Github veröffentlicht werden, werden die Benötigten Softwarepakete nicht mit hochgeladen. Damit die Software nach dem herunterladen funktioniert, müssen diese Pakete aus dem npm Repository heruntergeladen werden. Damit diese nicht von Hand heruntergeladen werden müssen, lädt npm automatisch die benötigten Pakete herunter. (siehe 3.2.3) [6]

2.7 node.js  
node.js ist eine Open-Source-Plattform, die zur serverseitigen Ausführung von JavaScript genutzt wird. node.js nutzt die freie JavaScript Implementierung V8, die in C/C++ geschrieben ist und somit eine schnelle Ausführung erlaubt. Der JavaScript-Code wird vor der Ausführung in nativen Maschinencode kompiliert. [7]

2.8 ReactJS  
ReactJs ist eine von Facebook entwickelte und unter der BSD-Lizenz veröffentlichte JavaScript-Bibliothek für Benutzeroberflächen (UI siehe 2.1). Es wird unter anderem von Facebook, Instagram, Whatsapp, Yahoo, AirBnb und vielen weiteren Firmen verwendet. Eine umfangreiche Liste ist unter folgendem Link zu finden: https://github.com/facebook/react/wiki/Sites-Using-React

ReactJS benutzt einen virtuellen DOM, was eine schnelle Renderingperformance zur Folge hat. Motivation von Facebook ReactJS zu entwickeln war besser verstehbaren und leicht zu wartenden Frontendcode zu produzieren. Deshalb wird auf eine modulare Komponentenarchitektur gesetzt. Desweiteren gewährleistet diese Bibliothekt Browserkompatibilität durch synthetische Events, um Browserunterschiede wegzuabstrahieren. [8]

2.9 JSX  
JSX ist eine Syntaxerweiterung zu JavaScript. Mit JSX wird beschrieben, wie die UI aussehen soll. In JSX können auch JavaScript Ausdrücke verarbeitet werden. Dazu werden diese in geschweifte Klammern geschrieben. Nach der Kompilierung werden JSX Ausdrücke in reguläre JavaScript Objekte umgewandelt.

## 2.10 Webpack

Mit Webpack können Module einer JavaScript Anwendung gebündelt werden. Webpack analysiert dabei alle Abhängigkeiten zu weiteren Modulen und fügt diese zu einer einzigen „vollständigen“ JavaScript Datei zusammen. Dann muss vom Browser nur noch eine einzige Datei verarbeitet werden. Speziell für die Entwicklung wird von Webpack der Webpack Dev Server zu Verfügung gestellt. Dabei handelt es sich um einen Webserver, der die entwickelte Anwendung ausliefert. Dieser Webserver stellt einen „Hot“-Mode zu Verfügung, mit der der Server automatisch Änderungen am Quellcode erkennt und diese dann nachlädt. So können ohne Aktualisierung des Browsers unmittelbar die durchgeführten Änderungen angeschaut werden. Da Webpack nur die veränderten Dateien neu übersetzt, geht die zudem sehr schnell. [9] [10]

### 2.10.1 Konkretes Ablaufbeispiel am verwendeten react-seed



Abbildung : Webpack

Durch den entry Eintrag in der webpack.config.js wird der Einstiegspunkt für Webpack definiert. Ab diesem Einstiegspunkt werden alle Abhängigkeiten zu anderen Modulen rekursiv aufgelöst. Durch den output Eintrag wird angegeben an welchem Speicherort die Ausgabe gespeichert wird. Im Falle des verwendeten react-seeds (siehe 3.1) sind die entry und output Einträge so definiert. Bei anderen Projekten kann dies beliebig anders gestaltet werden. Abbildung 1 soll den Ablauf verdeutlichen.

2.11 Git/GitHub  
Git ist eine freie Software zur Versionskontrolle bei Softwareprojekten. Git wurde im Jahr 2005 von Linus Torvalds entwickelt. Bei GitHub können kostenlose Online Repositories erstellt werden, um eigene Softwareprojekte mittels Git auf GitHub zu speichern. Der Vorteil der Online Speicherung ist, dass man die Software von jedem Computer mit Internetzugang herunterladen kann. Im Tutorial werde ich näher darauf eingehen wie ein Repository auf GitHub erstellt wird und wie mit Git gearbeitet wird. [11]

# 

# 3. Grundlagen

3.1 rect-seed als Grundgerüst  
Grundlage dieses Projekts bildet der react-seed von Matijas Milos (Benutzername Github: MatiMil), der als Gitub Repository unter folgendem Link bereitgestellt wird:

https://github.com/MatiMil/react-seed

Um mit dem react-seed zu arbeiten, kann dieser direkt von Github gedownloadet werden oder auch mittels GIT heruntergeladen werden. Dazu wird in der Windows/Linux/MAC-Konsole folgender Befehl eingegeben.

git clone https://github.com/MatiMil/react-seed.git

Zur Datensicherung empfiehlt es sich, für das eigene Projekt ebenfalls auf Github ein Repository anzulegen. Unter folgendem Link gibt es eine Schritt für Schritt Anleitung, wie ein neues Repository angelegt wird:

https://help.github.com/articles/create-a-repo/

Der react-seed kann als Grundgerüst für beliebige ReactJS Projekte genutzt werden. Das Grundgerüst gibt vor, wie die selbst erstellten Dateien sinnvoll gegliedert werden können. Dies ist nicht zwingend notwendig, aber es bietet eine Struktur die alle erstellten Projekte gleich gliedert. Somit wird die Wartbarkeit und Übersichtlichkeit der Projektstruktur gewährleistet. Der react-seed hat bereits eine Navigationsleite integriert, die auf mobilen Geräten in ein Hamburger Menü umgewandelt wird.

3.2 Einrichten der Projektumgebung

### **3.2.1 Übersicht** Installation git:

Schritt 1: GIT kann unter folgendem Link heruntergeladen werden:

https://git-for-windows.github.io/

Schritt 2: Auf folgender Webseite werden die Grundlegenden Befehle, um GIT zu benutzen erläutert:

https://rogerdudler.github.io/git-guide/index.de.html

Dieser kleine Einblick ist für dieses Projekt vollkommen ausreichend.

### **3.2.2 Installation Node.js LTS**

Schritt 1: node.js kann auf folgender Seite heruntergeladen werden:

https://nodejs.org/en/

Es empfiehlt sich die v6.10.3 LTS Version herunterzuladen und zu installieren. Bei der LTS Version handelt es sich um die Long Term Support Version, die über eine längere Zeit Unterstützt wird. Wenn immer die neuste Version installiert wird, in der neue Features hinzugekommen sind, kann es vorkommen, dass bereits erstellter Code nicht mehr funktioniert. Deshalb sollte die Version während der Entwicklung nicht geändert werden.

3.2.3 Installation der Abhängigkeiten  
Um die Anwendung starten zu können müssen zuerst alle Abhängigkeiten sprich alle benötigten Module heruntergeladen. Dazu muss auf der Konsole in das entsprechende Projekt gewechselt werden, in den der react-seed eingefügt wurde. Dann wird folgender Befehl ausgeführt:

npm -i

Dadurch werden alle Module, die in der package.json unter dependencies gelistet werden automatisch installiert. Wenn die installierten Module weitere Abhängigkeiten zu anderen Modulen besitzen werden diese automatisch mitinstalliert. Es werden somit alle Module installiert, bis alle Abhängigkeiten aufgelöst sind. Die Installation von Hand, bis alle Abhängigkeiten installiert sind, wäre ein mühevoller Prozess. Auch Webpack (siehe 2.10) wird mitinstalliert.

3.2.4 Zugriff eines Smartphones über das Homenetz  
Um die erstellte Anwendung zu testen, empfiehlt es sich die ReactJS-Anwendung im Homenetz verfügbar zu machen. So kann bequem über das eigene Smartphone getestet werden.

Dazu wird in der package.json folgendes neues Skript eingetragen (die IP-Adresse muss durch die eigene ersetzt werden):

"start": "webpack-dev-server --content-base src --inline --hot --host 192.168.178.20 --port 8080"

Unter Host wird die IP-Adresse eingetragen, auf der die ReactJS Anwendung gehostet wird. Unter Port kann ein beliebiger nicht belegter Port angegeben werden.

Dieses Skript wird mit folgendem Befehl gestartet:

npm run start

3.2.5 Starten des Projekts

Nun wird in der Konsole in das Projektverzeichnis gewechselt und dort folgender Befehl zum Starten der Anwendung eingegeben:

npm run start

Im Browser auf dem PC oder auf dem Smartphone kann mit 192.168.178.20:8080 aufgerufen werden. Es reicht die Anwendung einmal zu starten. Bei Änderungen am Quellcode, werden diese sofort Kompiliert und im Browser angezeigt. Die Entwicklung gestaltet sich so sehr angenehm, da die Änderungen automatisch ohne weiteres im Browser angezeigt werden. Wenn ein Syntax-Fehler besteht wird dies im Browser und in der Konsole, mit der entsprechenden Zeile angezeigt.

3.2.6 Erlernen der Grundlagen

Zum Erlernen der Grundlagen kann das von Facebook bereitgestellte Tutorial und Dokumentation verwendet werden.

https://facebook.github.io/react/

Außerdem gibt es ein interaktives Tutorial auf Codecademy:

https://www.codecademy.com/learn/react-101

Bei diesem Tutorial kann der Quellcode direkt im Browser eingegeben werden. Das Tutorial kontrolliert die Eingabe und wenn eine Aufgabe erfolgreich bearbeitet wurde kann die nächste Aufgabe erledigt werden.

### 3.2.7 React Developer Tools

React Developer Tools sind als Browserextension für Google Chrome sowie für Mozilla Firefox erhältlich. Diese können kostenlos im Chrome Web Store bzw. bei den Add-ons von Mozilla heruntergeladen werden (https://addons.mozilla.org/de/firefox/addon/react-devtools/). Desweiteren existiert eine Standalone App, die es ermöglicht mit anderen Browsern (Safari, Internet Explorer) und ReactNative zu debuggen. Die React Developer Tools erlauben es die Komponetnen Hirarchie, die props und die states zu überwachen. [12]

Unter folgendem Link gibt es eine genaue Erklärung wie die Developer Tools verwendet werden.

https://github.com/facebook/react-devtools

# 4. Umsetzung

### 4.1 Ziel des Informatikprojekts

Ziel des Informatikprojekts war es ein UI-Element zu entwickeln. Bei dem UI-Element handelt es sich um einen Button, der wenn er geklickt wird mehrere vordefinierte Auswahlmöglichkeiten bietet. Die Auswahlmöglichkeiten werden kreisrund um den mittleren Button angeordnet. Die Anzahl der Auswahlmöglichkeiten kann selbst gewählt werden. Es ist aber zu bedenken, dass bei zu vielen Auswahlmöglichkeiten die Übersichtlichkeit verloren geht. Als Erfahrungswert kann gesagt werden, dass deshalb nicht mehr als acht Auswahlmöglichkeiten zur Verfügung stehen sollten. In Abbildung 2 wird ein solcher Button abgebildet. Mit dem ersten Button „body color“ kann die Hintergrundfarbe der Webapp geändert werden. Der Button wird im weiteren Verlauf der Ausarbeitung Revolver Button genannt, da er, wenn er ausgeklappt ist aussieht wie die Trommel eines Revolvers.

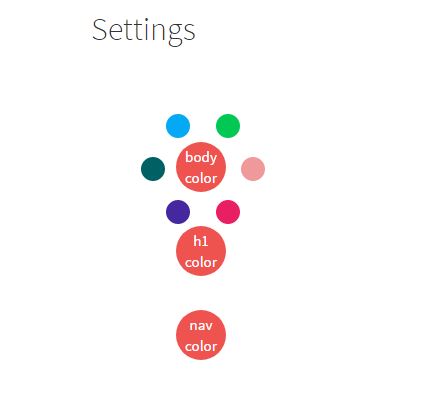


Abbildung : Button zum Ändern des Styles

### 4.2 Schrittweise Umsetzung von einem normalen Button zu einem fertigen Revolver Button

Auf github steht der komplette Quellcode des Projektes unter folgendem Link zur Verfügung:

https://github.com/achim1234/reactjs\_ui\_tutorial

In diesem Projekt unter dem Pfad src/js/compontents/tutorial steht für jeden Entwicklungsschritt ein Ordner bereit der den kompletten Code des jeweiligen Entwicklungsschritts bereitstellt. Außerdem wurde für jeden Entwicklungsschritt eine eigene Seite in dem Tutorial angelegt, der das Endergebnis

des aktuellen Schritts im Browser anzeigt. Somit kann das eigene Ergebnis sofort mit dem Sollergebnis abgeglichen werden.

**Schritt 1: Ausgeben eines normalen Buttons**

Zuerst wird der Entwicklungsserver wie in Punkt 3.2.3 gestartet. Im Browser wird dann die in dem Startskript vergebene IP-Adresse in folgendem Format eingegeben: xxx.xxx.xxx.xxx:port. In dem Verzeichnis reactjs\_webapp/src/js/components/tutorial/step1 wird ein neuer Ordner mit dem Namen RevolverButton erstellt. Darin wird eine Datei mit dem Namen Button.js angelegt. Um nun einen einfachen Button zu rendern muss zunächst eine neue Klasse angelegt werden. Diese wird gleich benannt wie der Dateiname (siehe Abbildung 3, Zeile 3). Mit den Schlüsselwörtern *export default* wird das Button Modul exportiert. Wenn diese Schlüsselwörter vorangestellt werden, kann das exportierte Modul in einem anderen Modul aufgerufen werden. In Abbildung 3, Zeile 2 wird der exportierte Button nun importiert. Der Name des importierten Moduls kann frei vergeben werden. Nach dem Schlüsselwort from wird der Pfad zu der Klasse angegeben, die importiert werden soll. In Zeile 10 wird der importierte Button mit <Button /> gerendert.

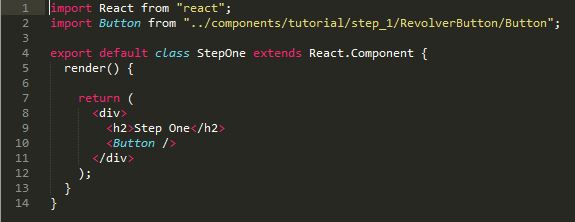


Abbildung : Ausgeben des Buttons auf der StepOne.js Page (src/js/pages/StepOne.js)

In Abbildung 4 wird in Zeile 11 ein Button zurückgegeben. Dieser wird immer dann zurückgegeben wenn wie in Abbildung 3, Zeile 2 der Button importiert und wie in Zeile 10 gerendert wird. Damit mehrzeiliger HTML-Code zurückgegeben werden kann, müssen nach dem *return*-Schlüsselwort runde Klammern folgen. Zudem muss der HTML-Code von einem div-Element umschlossen werden.

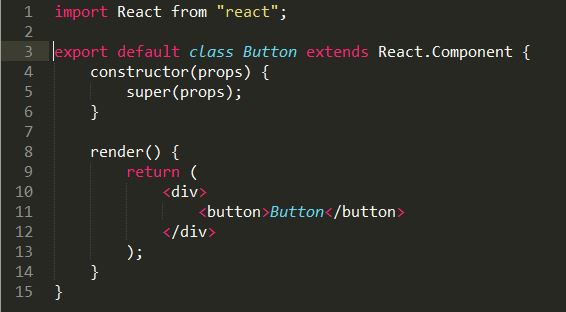


Abbildung 4: src/js/components/step\_1/tutorial/RevolverButton/Button.js

In Abbildung 5 kann die somit erzielte Ausgabe betrachtet werden.

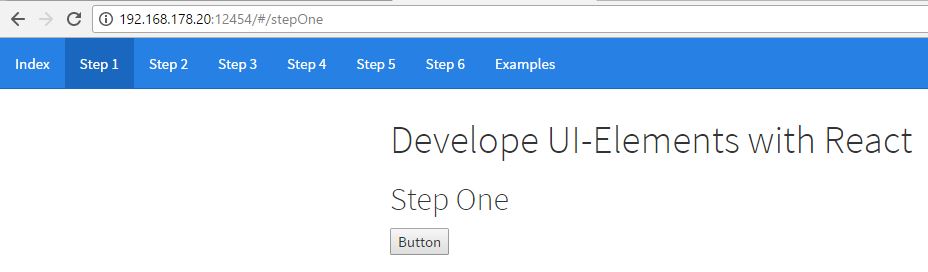


Abbildung : Ansicht Schritt 1 im Browser

**Schritt 2: Anpassen des Stylings**

Da sich für den RevolverButton ein kreisförmiges Design der Buttons herauskristallisiert hat, wird in diesem Schritt gezeigt wie das Styling des Buttons angepasst werden muss, um eine kreisförmige Erscheinung zu erhalten. Das Styling erfolgt über CSS-Anweisungen. Zu beachten ist das die CSS-Anweisungen nicht wie gewohnt mit Bindestrich z.B. *background-color*, sondern in lowerCamelCase-Schreibweise ohne Bindestrich geschrieben werden z.B. *backgroundColor*.

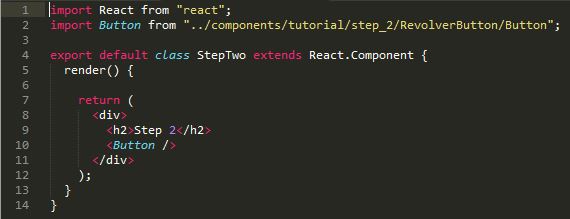


Abbildung : Ausgeben des Buttons auf der StepTwo.js Page (src/js/pages/StepTwo.js)

Die Styling Angaben werden in Abbildung 7 von Zeile 9 bis 16 vor dem *return*-Statement in einer konstanten Variablen definiert. In Zeile 20 wird die Variable *mainButton* in geschwungenen Klammern an das *style*-Attribut des Buttons übergeben. In geschwungenen Klammern wird immer JavaScript Code ausgeführt. Um in CSS einen Kreis zu erhalten müssen die *width* und die *height* auf die gleiche Größe gesetzt werden. Der *borderRadius* muss genau die Hälfte dieses Wertes betragen. Dass es sich um px („Pixel“) handelt muss nicht extra angegeben werden, dies wird automatisch angenommen wenn nur eine Zahl angegeben wird.



Abbildung : Schritt 2: Styling des Buttons

In Abbildung 8 kann die somit erzielte Ausgabe betrachtet werden.

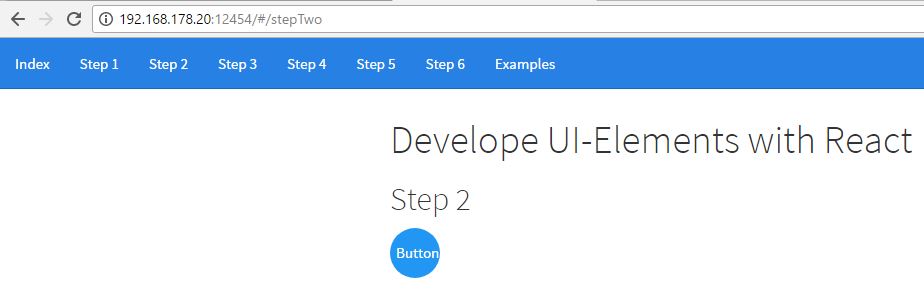


Abbildung : Ansicht Schritt 2 im Browser

**Schritt 3: Revolver Button Items hinzufügen**

In diesem Schritt werden die Items hinzugefügt, die beim Öffnen des Buttons angezeigt werden sollen. Bei den Items handelt es sich ebenfalls um Buttons. Für diese Items wird eine neue Klasse RevolverItem in dem Verzeichnis src/js/components/tutorial/step\_3/RevolverButton angelegt.

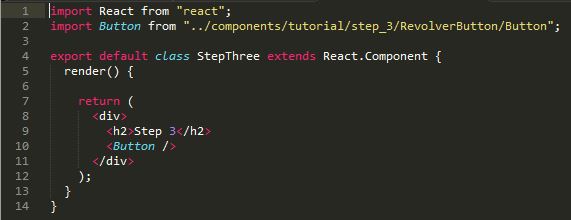


Abbildung : Ausgeben des Buttons auf der StepThree.js Page (src/js/pages/StepThree.js)

Abbildung 10: Beschreibung des Quellcodes

In Zeile 7-9 wird ein Status showComponent gesetzt. Wenn die Klasse aufgerufen wird, wird der Status im Konstruktor auf false gesetzt. Dieser Status wird dazu verwendet um die RevolverItems nicht anzuzeigen bzw. nach der Änderung des Status auf true anzuzeigen. Beim Start der Anwendung werden die RevolverItems somit nicht angezeigt.

In Zeile 10 wird ein Binding durchgeführt. Dies ist nötig um in dem toogleShowHide-Listener „this“ verwenden zu können. Dieser Listener wird immer dann aufgerufen wenn der Button getoucht bzw. gecklickt wird.

In Zeile 14 wird ein neuer Status in der Variable newState abgespeichert. Mit this.state.showComponent == wird der aktuelle Status verglichen. Wenn der Status false ist, wird er auf true gesetzt und umgekehrt. Dies wird durch den ?-Operator bewerkstelligt. Mit der Funktion this.setState wird der neue Status gesetzt.

In Zeile 42-45 wird ein ähnliches Konstrukt wie zuvor beschrieben genutzt. Wenn this.state.showComponent auf true gesetzt ist wird der RevolverItem gerendert, wenn nicht wird die Komponente auf null gesetzt. Somit wird nichts ausgegeben.

In Zeile 43 werden folgende Attribute an die Instanz der Komponente übergeben: value und styleProp. Über diese Attribute können Werte von einer Elternkomponente zu einer Kindkomponente übergeben werden. In der Kindkomponente RevolverItem kann über this.props.value bzw. this.props.styleProp auf die übergebenen Werte zugegriffen werden. Dies wird in Abbildung 11 Zeile 10 verwendet um den Namen des Buttons zu bestimmen und der Style der angewendet werden soll wird gesetzt.

In Zeile 28-35 wurde das Styling für die RevolverItems definiert. Sie erhalten auch ein kreisrundes Aussehen, sowie einen Zahlenwert der durch Auswahl einer bestimmten Zahl ausgewählt wird. In diesem Entwicklungsschritt sind die RevolverItems noch unterhalb des Buttons platziert. Im nächsten Entwicklungsschritt werden diese kreisrund um den Button Plaziert.



Abbildung 10: src/js/components/step\_3/RevolverButton/Button.js

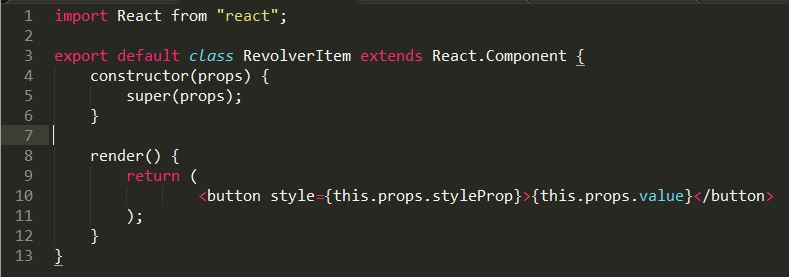


Abbildung 11: src/js/components/step\_3/tutorial/RevolverButton/RevolverItem.js

In Abbildung 12 kann die somit erzielte Ausgabe betrachtet werden.



Abbildung : Ansicht Schritt 3 im Browser

**Schritt 4: Kreisförmige Anordnung der Items**

In diesem Schritt werden die RevolverItems kreisrund um den Button platziert. Dies ermöglicht ein elegantes Design sowie eine übersichtliche Auswahlmöglichkeit. Zur Berechnung der kreisförmigen Anordnung wird ein fertiger Codeschnipsel von folgender Webseite verwendet:

https://codepen.io/anon/pen/vKpPBJ

Der Code wurde leicht abgeändert, um diesen an die Bedürfnisse des RevolverButtons anzupassen.

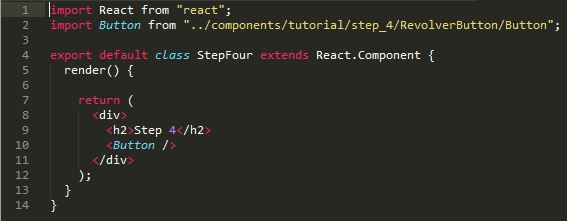


Abbildung : Ausgeben des Buttons auf der StepFour.js Page (src/js/pages/StepFour.js)

In Abbildung 14 wurden nur Änderungen der render-Methode vorgenommen. Das styleProp Attribut wird entfernt. Der Style wird jetzt in dem RevolverItem direkt gesetzt. Damit die Formel zur kreisrunden Anordnung Funktioniert werden zwei neue Attribute currentItem und maxItems benötigt. Bei currentItem wird eine fortlaufende Nummer, bei 1 beginnend vergeben. maxItems gibt an, wie viele RevolverItems insgesamt gerendert werden.

In Zeile 31-35 wird das Umschliesende div-Element in Zeile 38 als inline-Block dargestellt, mit einem marginLeft von 70 Pixeln. Das heißt es können mehrere Buttons nebeneinander mit einem Abstand von 70 Pixeln nach links dargestellt werden.

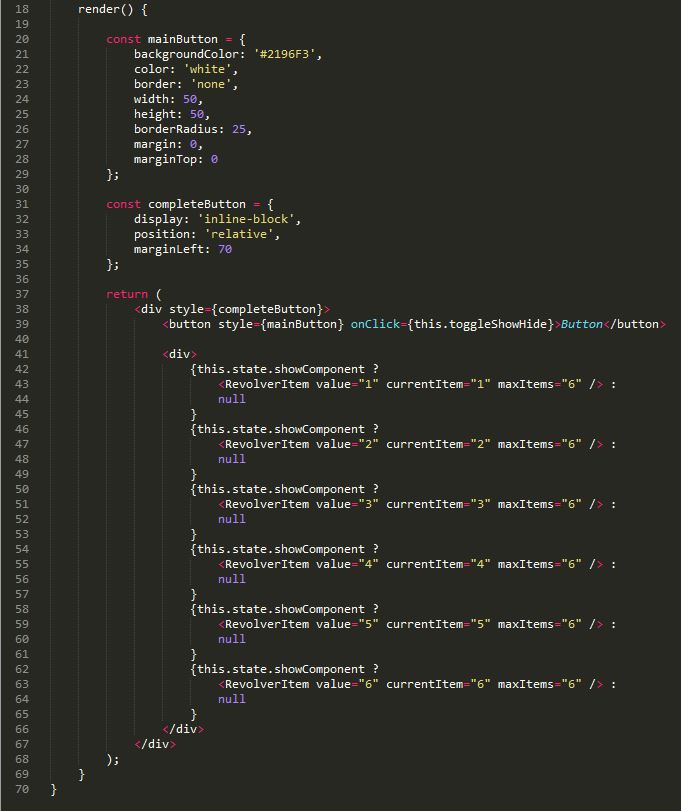


Abbildung 14: src/js/components/step\_4/tutorial/RevolverButton/Button.js

In Abbildung 15 wird vor dem *return*-Statement die x und y Werte für die kreisrunde Anordnung berechnet. Über die Variable radius kann angegeben werden in welchem radius die Elemente angeordnet werden können. Über die offsetX und offsetY Werte kann die Positionierung der gesamt erzeugten RevolverItems um die x bzw. y-Achse verschoben. Hier werden sie so gewählt das der Button mittig in den RevolverItems platziert wird. Dann werden die berechneten Werte in Zeile 29-30 an die Styling Variable übergeben. Durch die Absoolute-Positionierung verschieben sich die anderen HTML-Elemente die sich auf der Seite befinden können nicht.



Abbildung 15 : src/js/components/step\_4/tutorial/RevolverButton/RevolverItem.js

In Abbildung 16 kann die somit erzielte Ausgabe betrachtet werden.



Abbildung : Ansicht Schritt 4 im Browser

**Schritt 5: Flexibilität des Buttons über Attribute gewährleisten**

Da zum jetzigen Zeitpunkt alle Parameter wie z.B. die Anzahl der RevolverItems, den Wert den sie Anzeigen usw. nur durch Änderungen im Quellcode geändert werden können wird der Button in diesem Entwicklungsschritt über Attribute flexibel gestaltet. So kann der Button für unterschiedliche Einsatzzwecke genutzt werden, ohne dass Änderungen im Quellcode erfolgen müssen. Das erleichtert den Umgang mit dem RevolverButton enorm. So muss der Programmierer, der den RevolverButton einsetzt nur in der Dokumentation nachschauen welche Attribute zur Verfügung stehen und wie diese befüllt werden müssen. Der RevolverButton besitzt nun die Attribute btnName, maxItems und values. Über btnName wird ein Name des Buttons vergeben, der so im Browser angezeigt wird. maxItems legt fest wieviele RevolverItems ausgegeben werden sollen. Über das Attribut values wird ein Array mit den Werten die die RevolverItems haben sollen übergeben.

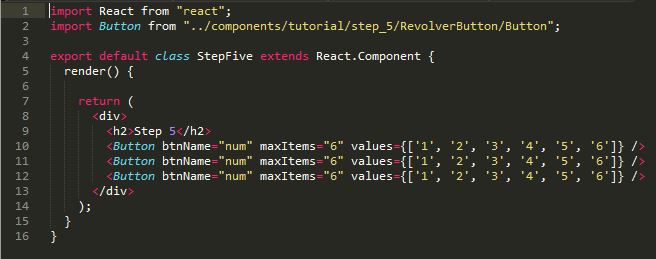


Abbildung : Ausgeben des Buttons auf der StepFive.js Page (src/js/pages/StepFive.js)

Beschreibung Quellcode Abbildung 18:

In Zeile 50 wird ein leeres Array RevolverItems angelegt. Dieses wird später die Anzahl an RevolverItems enthalten die über das Attribut maxItems in Abbildung 17 Zeile 10-12 vergeben wurden.

In Zeile 51-54 werden in einer Schleife die benötigten RevolverItems instanziiert und in das Array gespeichert. Dies geschieht in Zeile 52-53 über die push()-Methode. Außerdem werden in Zeile 53 die in der RevolverItems-Klasse benötigten Attribute übergeben. Die so angelegten RevolverItems werden in Zeile 59 mit {revolverItems} ausgegeben. In Zeile 58 wird für den Button ein onClick sowie ein onTouchCancel Listener verwendet. Somit kann der Button auf Pcs und auf Smartphones bedient werden.

In Zeile 53 wird die toggleShowHide-Methode an das RevolverItem übergeben.

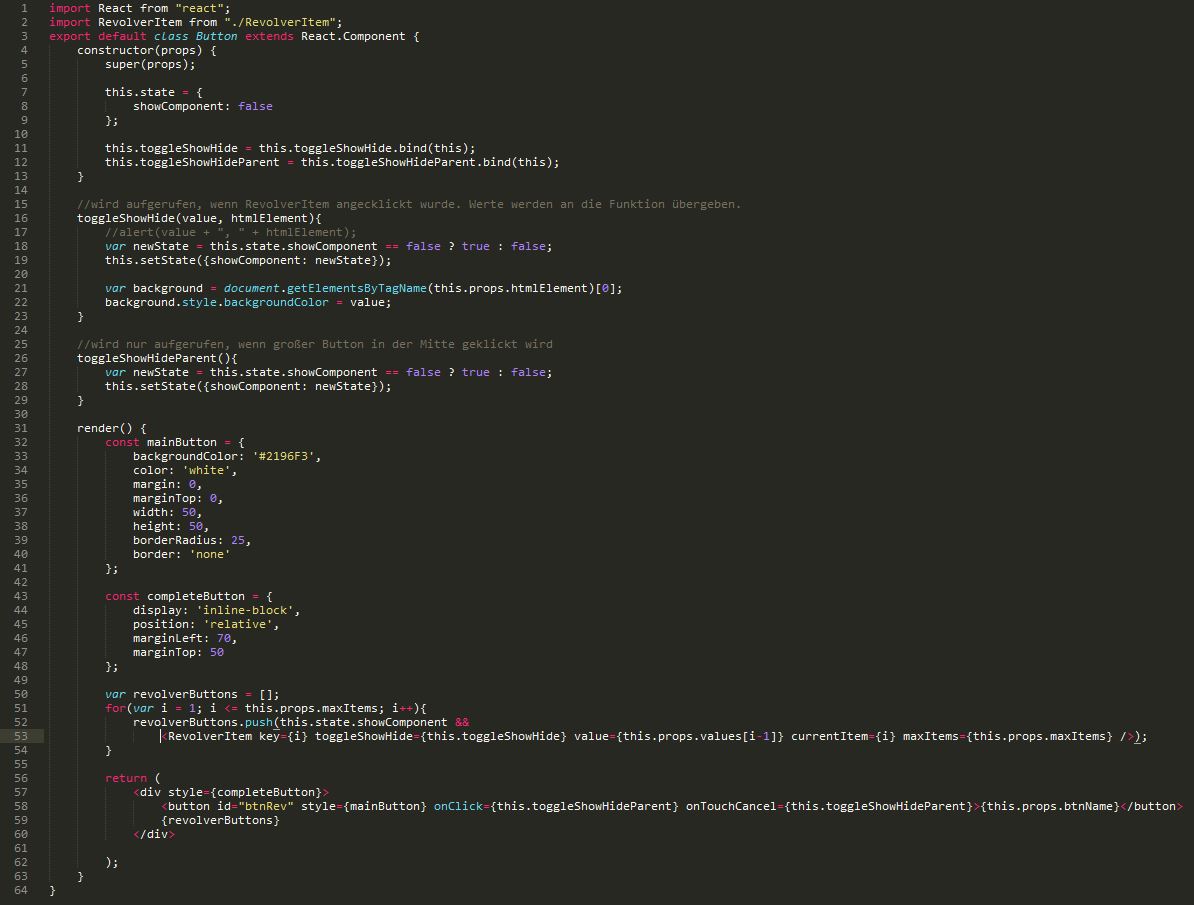


Abbildung 18: src/js/components/tutorial/step\_5/RevolverButton/Button.js

In Zeile 33 in Abbildung 19 wird die toggleShowHide-Methode aus den props übernommen und in einer Variablen gespeichert. In dem onClick bzw. onTouchCancel Listener in Zeile 36 bzw. 37 kann somit die toggleShowHide-Methode des Buttons aufgerufen werden. Es werden der value und das htmlElement übergeben (wird erst im nächsten Schritt benötigt). Aber in diesem Schritt wird lediglich der Status showComponent wieder auf false gesetzt und die RevolverItems werden wieder verborgen.

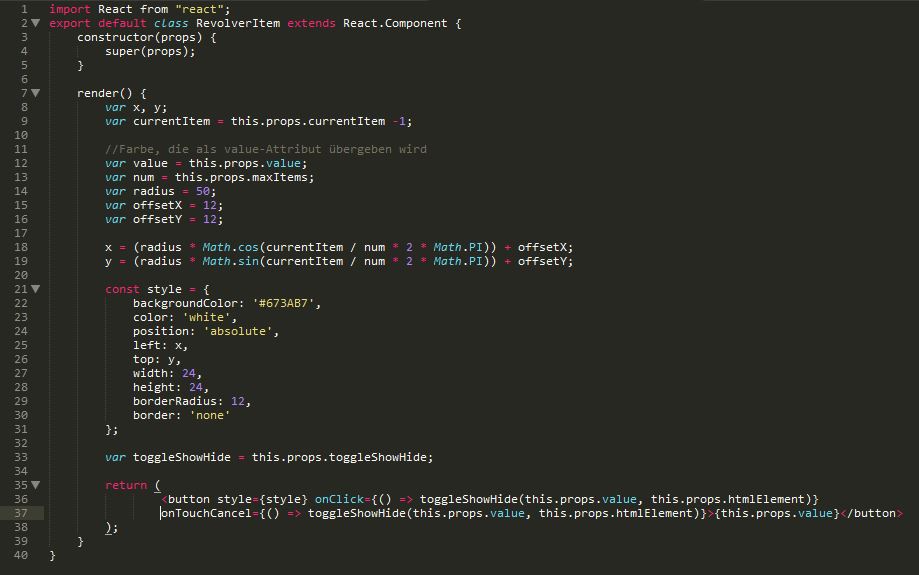


Abbildung 19: src/js/components/step\_5/RevolverButton/RevolverItem.js

In Abbildung 20 kann die somit erzielte Ausgabe betrachtet werden.

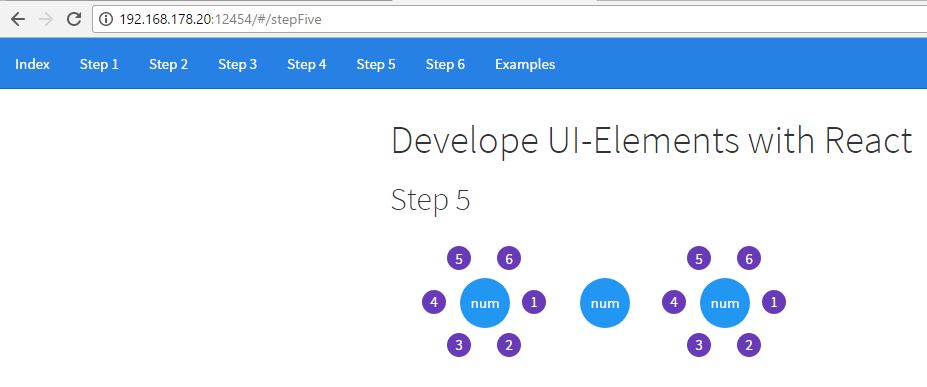


Abbildung : Ausgabe Schritt 5 im Browser

**Schritt 6:** Im letzten Schritt wird der RevolverButton so angepasst, dass damit das Styling der Webapp angepasst werden kann. Beispielsweise kann die Farbe der Navigationsleiste, die Hintergrundfarbe usw. geändert werden. Die Funktion ist die gleiche wie in Schritt 5. Hier kann zusätzlich noch ein HTML-Element angegeben werden, dessen Style geändert werden soll.

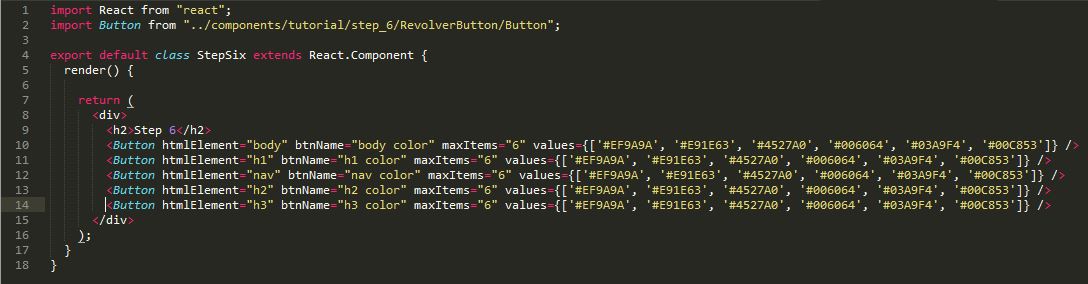


Abbildung : Ausgeben des Buttons auf der StepSix.js Page (src/js/pages/StepSix.js)

In der toggleShowHide-Methode in Zeile 16 werden der value und das htmlElement des angeklickten RevolverButtons übergeben. In Zeile 20 wird jeweils das erste HTML-Element der Seite ausgewählt und in Zeile 21 wird der übergebene Farbwert gesetzt.

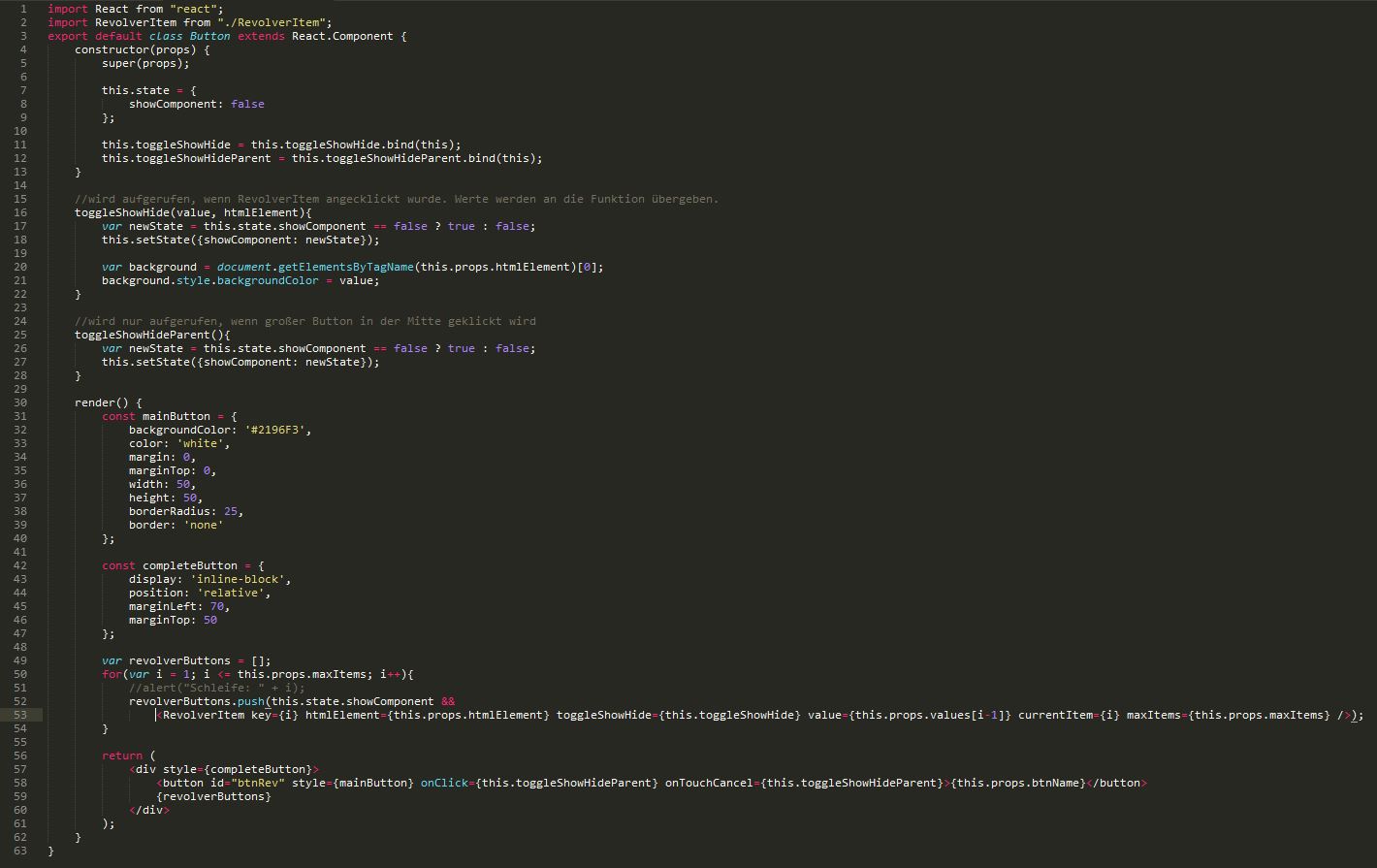


Abbildung 22: src/js/components/step\_6/RevolverButton/Button.js

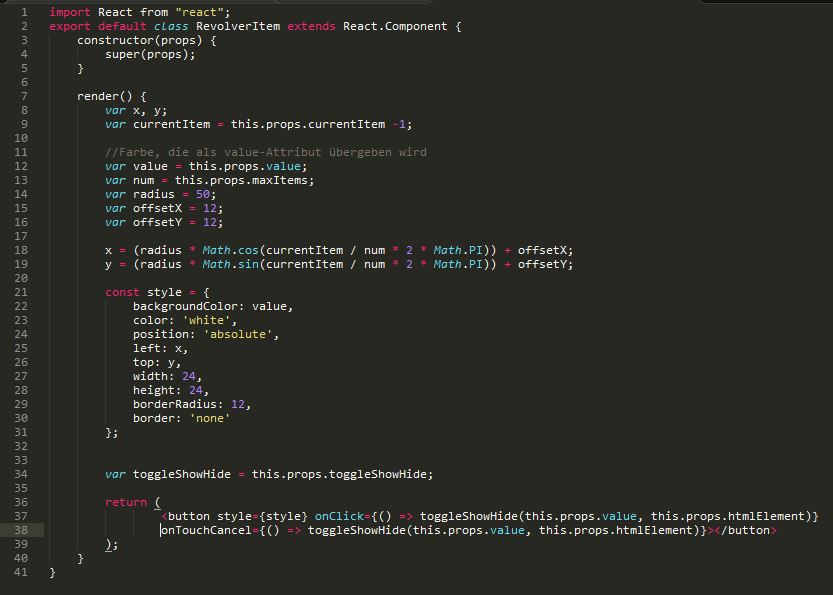


Abbildung 23: src/js/components/step\_6/RevolverButton/RevolverItem.js

In Abbildung 24 kann die somit erzielte Ausgabe betrachtet werden.

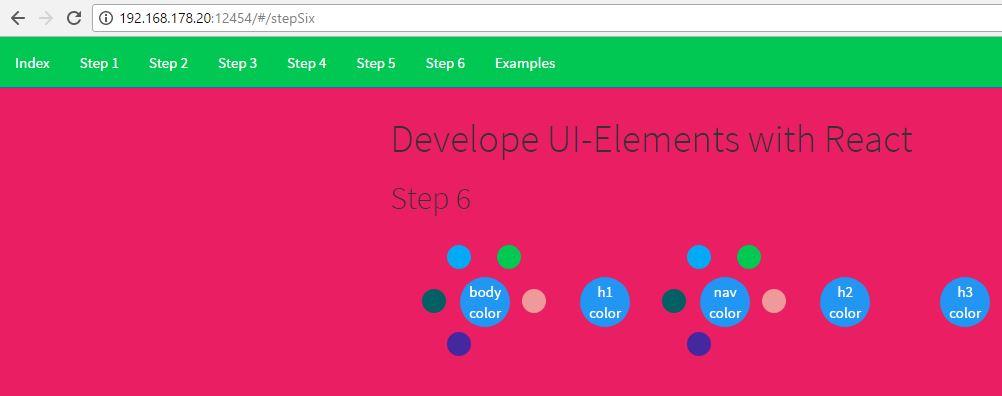


Abbildung : Ausgabe Schritt 6 im Browser

# 

# 5. Nutzertests

Um die User Experience (Benutzererfahrung) bei der Benutzung des RevolverButtons zu verbessern und Unterschiede zu einem Dropdown-Menü darzustellen wurde ein Thinking Aloud Test mit zwei fest definierten Use Cases mit zwei Versuchspersonen durchgeführt. Bei diesem Test werden verschiedene Handlungsszenarien erstellt, die von einem durchschnittlichen Benutzer beim Umgang mit dem RevolverButton erwartet werden. Der Test wurde jeweils am Desktop und Smartphone durchgeführt.  
Im ersten Use Case soll in einem simulierten Internetshop ein Kugelschreiber mit grüner Farbe bestellt werden.   
Im zweiten Use Case soll in einem simulierten Kraftstoffrechner eine Automarke und eine Kraftstoffart ausgewählt werden um den Kraftstoffverbrauch des gewählten Autos zu berechnen. Die Versuchspersonen erhalten Aufgaben, die sie Ausführen sollen. Dabei sollen sie ihre Gedanken laut aussprechen wie z.B. was sie bei der Verwendung des RevolverButtons denken, welche Fragen sie haben und was sie nicht verstehen. Im zweiten Teil sollen sie Unterschiede zwischen dem Dropdown-Menü und dem RevolverButton bennenen. Diese Äußerungen werden protokolliert und im Anschluss ausgewertet. Diese Ergebnisse können dann mit den von mir erwarteten Handlungen abgeglichen werden. Wichtig während des Tests ist es den Nutzern das Sprechen zu überlassen, um unverfälschte Ergebnisse zu erhalten.

**Versuchsperson 1 (Geschlecht: weiblich, Alter: 19 Jahre):**

**Use Case 1 Dropdown Desktop:**

-Annahme: Beim Klick auf den Pfeil des Dropdown-Menüs öffnet sich eine Farbauswahl, aus der die Farbe grün gewählt werden kann.

-Dropdownmenüs sind der Testerin bekannt.

**Use Case 1 RevolverButton Desktop:**

-Annahme: Beim Klick auf den RevolverButton öffnet sich eine Farbauswahl. Allerdings ist die Testerin sich unsicher wie die Farbauswahl gestaltet ist.

-Nachdem die Testerin auf den RevolverButton geklickt hat war, Sie einen Moment über die Anordnung der Elemente verblüfft, hat sich aber schnell zurechtgefunden und konnte die grüne Farbe auswählen. Auch sei die Rückmeldung über die ausgewählte Farbe gut, da diese dann zur Farbe des Hauptbuttons wird. Außerdem sei die Farbauswahl besser als die Farbe als Wort zu lesen. Der RevolverButton sei auch mit der Maus leicht zu bedienen.

**Use Case 1 Dropdown Smartphone:**

-Auch hier klappt die Farbauswahl ohne Probleme. Allerdings wäre eine Rückmeldung, welche Farbe ausgewählt wurde sinnvoll.

**Use Case 1 RevolverButton Smartphone:**

-Auch hier werden die gleichen Vorteile wie bei der Desktopvariante genannt. Die Bedienung mit den Fingern ist problemlos zu bewerkstelligen. Daraus folgt, dass die Elemente des RevolverButtons nicht zu klein dimensioniert sind.

**Fazit Use Case 1:**

Der Use Case 1 ließ sich mit beiden Bedienelementen problemlos bewerkstelligen. Dropdown-Menüs waren der Testerin bekannt. Obwol der RevolverButton nicht bekannt war, konnte er von der Testerin problemlos bedient werden. Der RevolverButton stellt die Farbauswahl übersichtlich dar und lässt sich intuitiv bedienen. Die Rückmeldung, welche Farbe ausgewählt wurde (über die Farbe des Hauptbuttons) ist positiv hervorzuheben.

**Use Case 2 Dropdown Desktop:**

-Die Auswahl der Automarke und einer Krafftstoffart geht einfach vonstatten. Auch sei die Auswahl bei einer größeren Anzahl von Auswahlelementen einfach und es kann nach dem Anfangsbuchstaben über die Tastatur die gewünschte Automarke bzw. Krafftstoffart gesucht werden.

**Use Case 2 RevolverButton Desktop:**

-Bei der Auswahl der Automarke ist die Testerin verwirrt, da neun Elemete angezeigt werden. Hier dauert die Auswahl deutlich länger als beim Dropdown-Menü.

-Bei der Kraftstoffart ist die Auswahl wieder übersichtlich und kann schneller ausgewählt werden.

**Use Case 2 Dropdown Smartphone:**

-Die Auswal geht hier schnell vonstatten.

**Use Case 2 RevolverButton Smartphone:**

-Auch hier werden die gleichen Vorteile wie bei der Desktopvariante genannt.

**Fazit Use Case 2:**

Bei einer größeren Auswahl ca. ab acht Elementen wird der RevolverButton unübersichtlich und verwirrt die Testerin, vor allem wenn die RevolverItems längere Strings enthalten. Bei der Krafftstoffauswahl mit nur sechs Elementen fällt der Testerin die Auswahl leichter. Ab acht Elementen schneidet das Dropdown-Menü besser ab. Es verwirrt die Testerin nicht und die Auswahl kann schneller ausgewählt werden.

**Versuchsperson 2 (Geschlecht: männlich, Alter: 19 Jahre):**

**Use Case 1 Dropdown Desktop:**

-Der blaue Hover-Effekt sei störend und die Auswahl der Farben sei unübersichtlich.

**Use Case 1 RevolverButton Desktop:**

-Es war dem Tester sofort klar, dass er auf den Hauptbutton des Revolverbuttons klicken muss. Über die Anordnung der RevolverItems war der Tester kurz irritiert, konnte die Aufgabe die grüne Farbe für den Kugelschreiber auszuwählen aber ohne Probleme bewerkstelligen. Die Farbauswahl sei übersichtlicher als das Dropdown-Menü und es findet eine gute Rückmeldung über die ausgewählte Farbe statt.

**Use Case 1 Dropdown Smartphone:**

-Hier sei die Auswahl der Farbe übersichtlicher, als bei der Desktopvariante, da hier keine Farbe bei der Auswahl hinterlegt ist.

**Use Case 1 RevolverButton Smartphone:**

-Auch hier werden die gleichen Vorteile wie bei der Desktopvariante genannt.

**Fazit Use Case 1:**

Der RevolverButton hat hier deutlich besser als das Dropdown-Menü abgeschnitten. Die Auswahl über den RevolverButton ist übersichtlich und intuitiv.

**Use Case 2 Dropdown Desktop:**

-Auswahl sei übersichtlich und schnell.

**Use Case 2 RevolverButton Desktop:**

-Auswahl der Automarke ist verwirrend und unübersichtlich. Die Auswahl der Krafftstoffart ist übersichtlich, aber das Dropdown-Menü sei trotzdem übersichtlicher.

**Use Case 2 Dropdown Smartphone:**

-Auch hier erfolgt die Auswahl schnell und zielgerichtet.

**Use Case 2 RevolverButton Smartphone:**

-Auch hier ist das Ergebnis wie bei der Desktopvariante.

**Fazit Use Case 2:**

Hier schneidet das Dropdown-Menü besser ab. Hier ist die Auswahl bei vielen Elementen deutlich einfacher. Der RevolverButton hat in diesem Use Case den Tester eher verwirrt.

**Fazit Gesamt (Dropdown-Menü vs. RevolverButton):**

Der RevolverButton schneidet bei der Farbauswahl deutlich besser als das Dropdown-Menü ab. Die Testpersonen haben sich, obwohl sie den RevolverButton noch nie bedient haben, recht schnell zurechtgekommen. Außerdem sei der RevolverButton optisch ansprechender als das Dropdown-Menü und die Rückmeldung der ausgewählten Farbe im Hauptbutton haben alle Tester positiv empfunden.   
Wenn sich aber zu viele Elemente (bei den Nutzertests etwa ab acht Stück) wurde die Auswahl für alle Testpersonen zu verwirrend und waren kurzzeitig überfordert eine schnelle Auswahl zu treffen. Hier haben die Dropdown-Menüs besser abgeschnitten. Außerdem kann bei der Desktopvariante nach den Anfangsbuchstaben der Auswahl gesucht werden. Um die Übersichtlichkeit bei dem RevolverButton zu gewährleisten sollten nach den Ergebnissen nicht mehr als acht Elemente pro RevolverButton verwendet werden. Es sollte darauf geachtet werden die Beschriftungen so kurz wie möglich zu halten. Deshalb ist der RevolverButton nur bei bestimmten Szenarien die die zuvor beschriebenen Anforderungen erfüllen sinnvoll einzusetzen. Hier kann der RevolverButton einen Mehrwert, gegenüber Dropdown-Menüs schaffen.

**Weitere Einsatzmöglichkeiten:**

Weitere Einsatzmöglichkeiten auch in Bezug auf IoT (Internet of Things) wäre z.B. die Dimmung der Beleuchtung auf einige vordefinierte Helligkeitsstufen, die Steuerung von Rolladen z.B. ganz geöffnet, ganz geschlossen, halb geöffnet usw.

# Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | [Online]. Available: http://www.itwissen.info/Benutzeroberflaeche-user-interface-UI.html. [Zugriff am 17 Mai 2017]. |
| [2] | 17 Februar 2017. [Online]. Available: https://de.wikipedia.org/wiki/Cascading\_Style\_Sheets. [Zugriff am 8 März 2017]. |
| [3] | „Wikipedia,“ 24 Februar 2017. [Online]. Available: https://de.wikipedia.org/wiki/Bootstrap\_(Framework). [Zugriff am 8 März 2017]. |
| [4] | 24 Februar 2017. [Online]. Available: https://de.wikipedia.org/wiki/JavaScript. [Zugriff am 8 März 2017]. |
| [5] | „Wikipedia,“ 8 Dezember 2016. [Online]. Available: https://de.wikipedia.org/wiki/Integrierte\_Entwicklungsumgebung. [Zugriff am 13 März 2017]. |
| [6] | „Wikipedia,“ 16 Januar 2017. [Online]. Available: https://de.wikipedia.org/wiki/Node\_Package\_Manager . [Zugriff am 16 Mai 2017]. |
| [7] | „Nodecode,“ 8 Oktober 2013. [Online]. Available: http://nodecode.de/was-ist-nodejs. [Zugriff am 23 Mai 2017]. |
| [8] | [Online]. Available: http://reactjs.de/posts/react-tutorial. [Zugriff am 17 Mai 2017]. |
| [9] | [Online]. Available: http://reactjs.de/posts/react-anwendungen-mit-webpack-entwickeln. [Zugriff am 27 Mai 2017]. |
| [10] | [Online]. Available: https://webpack.js.org/concepts/. [Zugriff am 27 Mai 2017]. |
| [11] | [Online]. Available: https://www.atlassian.com/git/tutorials/what-is-git. [Zugriff am 17 Mai 2017]. |
| [12] | [Online]. Available: https://github.com/facebook/react-devtools. [Zugriff am 16 Mai 2017]. |

## Selbständigkeitserklärung

Ich versichere, dass ich diesen Praktikumsbericht selbständig verfasst und keine anderen als  
die angegebenen Quellen und Hilfsmittel benutzt habe.  
Die Abbildungen in diesem Bericht sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Jettkofen, den 6. Juni 2017

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Datum, Ort, Achim Strohm