

Fachbereich VI  
Bachelor of Science  
Studiengang: Medieninformatik

Bachelorarbeit

Zur Erlangung des akademischen Grades  
„Bachelor of Science“ im Fach  
Medieninformatik

Erstellung einer mobilen und webbasierten verteilten Anwendung zum synchronen Datenaustausch von taktischen Spielzügen zwischen Clients auf Basis von spielrelevanten Inhalten

Betreuer:

Dr. Michael Steppat

Gutachter:

Prof. Dr. Stefan Edlich

Verfasser:

Robert Schierz  
Karl-Marx-Str. 5  
12043 Berlin  
Matrikelnr.: 798516

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die fremden Quellen wörtlich oder sinngemäß entnommen wurden, sind kenntlich gemacht. Die Arbeit war in gleicher oder ähnlicher Form noch nicht Bestandteil einer Studien- oder Prüfungsleistung.

Berlin, den 26.08.2015

|  |
| --- |
| Robert Schierz |

Abstract

Inhaltsverzeichnis

[Eidesstattliche Erklärung i](#_Toc443584469)

[Abstract ii](#_Toc443584470)

[1 Einleitung 1](#_Toc443584471)

[1.1 Thema 1](#_Toc443584472)

[1.1.1 Themenfindung 2](#_Toc443584473)

[1.1.2 Arbeitsteilung 2](#_Toc443584474)

[1.2 Motivation 2](#_Toc443584475)

[1.3 Zielsetzung 2](#_Toc443584476)

[2 Grundlagen 2](#_Toc443584477)

[2.1 Verwendete Frameworks 2](#_Toc443584478)

[2.1.1 JQuery 2](#_Toc443584479)

[2.1.2 JQueryUI 2](#_Toc443584480)

[2.1.3 SocketIO 2](#_Toc443584481)

[2.1.4 Bootstrap 2](#_Toc443584482)

[2.2 Verwendete Programmier und Strukturiersprachen 2](#_Toc443584483)

[2.2.1 Javascript 2](#_Toc443584484)

[2.2.2 HTML und CSS 2](#_Toc443584485)

[2.3 Datenformat 2](#_Toc443584486)

[2.3.1 JSON 2](#_Toc443584487)

[3 Konzeption 2](#_Toc443584488)

[3.1 Anforderungserhebung 2](#_Toc443584489)

[3.1.1 Klassendiagramm 2](#_Toc443584490)

[3.1.2 Usecaseanalyse 2](#_Toc443584491)

[3.2 Design 2](#_Toc443584492)

[3.2.1 Designkonzeption 2](#_Toc443584493)

[3.2.2 Benutzerführung 2](#_Toc443584494)

[3.2.3 Mokups 3](#_Toc443584495)

[3.3 Spezifikation 3](#_Toc443584496)

[4 Implementierung 3](#_Toc443584497)

[4.1 Entwurf 3](#_Toc443584498)

[5 Qualitätssicherung 3](#_Toc443584499)

[5.1 Funktionstests 3](#_Toc443584500)

[5.2 Usability-Test 3](#_Toc443584501)

[6 Fazit 3](#_Toc443584502)

[6.1 Ausblick 3](#_Toc443584503)

Einleitung

Thema

Themenfindung

Verteilte Anwendungen sind aus unserer heutigen Zeit nicht mehr wegzudenken. Sie ermöglichen den digitalen Informationsaustausch zwischen geografisch unabhängig befindlichen Endgeräten in einem Netzwerk. Als Informationsverteilungssystem fungiert in solch einer Anwendung ein Server. Dieser Server stellt sowohl die Persistenzschicht als auch die Vermittlungsschicht da. Die Empfänger digitaler Informationen einer verteilten Anwendung bezeichnet man als Clients. Die Clients stellen Endgeräte dar, welche in erster Linie Informationen des Servers anfordern. Die Persistenzschicht einer solchen Anwendung ermöglicht es, den Clients nicht nur Informationen anzufordern und zu verarbeiten, sondern die vom Anwender verarbeiteten Informationen zudem persistent zu speichern. Diese gespeicherten Daten werden innerhalb des Servers in einer Datenbank hinterlegt, und können nun jederzeit vom jeweiligen Client wieder angefordert werden. Dies ermöglicht dem Anwender, einen Ein- und Ausgabefluss an Informationen mit dem Server herzustellen.   
  
Ziel der Arbeit ist es eine Software zu entwickeln, welche es den Anwendern ermöglicht einen komplexen Gedankengang digital anschaulich aufzubereiten, und mit anderen Clients zu teilen. Es soll den Anwender möglich gemacht werden, eine spielrelevante Strategie auf eine in Vogelperspektive dargestellten Karte zu zeichnen. Der Anwender soll die Möglichkeit haben, diese Strategien effizient mit seinen Mitspielern teilen zu können. Aus der Sicht des Anwenders ist die Software in drei Ebenen untergliedert.  
  
Die erste Ebene beschäftigt sich damit, dass der Anwender erstellte Strategien, speichern, laden und löschen kann. Alle Strategien, die in der ersten Ebene erstellt wurden, können ausschließlich von dem Benutzer bearbeitet werden, der sie erstellt hat.  
  
Die zweite Ebene beinhaltet das Gruppensystem. Die Anwendung soll über ein Gruppensystem verfügen, mit welchem Gruppen erstellt werden können. Solch eine erstellte Gruppe soll die Mitglieder sowie die mit der Gruppe geteilten Strategien beinhalten. Entscheidet sich der Anwender eine Strategie zu speichern, so wird ihm die Möglichkeit gegeben, diese Strategie in eine Gruppe zu speichern. Die Strategie wird ausschließlich den Gruppenmitgliedern angezeigt und kann von diesen bearbeitet werden.  
  
Die dritte Ebene umfasst den Livemodus. Innerhalb dieses Livemodus können Anwender Strategien mit Gruppenmitgliedern in nahezu Echtzeit erstellen. Dabei soll es möglich sein, das jeder Teilnehmer des Livemodus, die Interaktionen der anderen Anwender mit der Strategie in nahezu Echtzeit sehen kann.

Das Computerspiel Counterstrike Global Offensive gehört zu dem Genre Online-Tactik-Shooter.  
Es wurde von den Spieleentwicklungsfirmen Valve und Hidden Path Entertainment im Jahre 2012 herausgebracht, und auf der mit Valve verbundenen Spielevertriebsplattform Steam vermarktet.  
Mittlerweile verbucht das Spiel stetig steigende Spielerzahlen, und gehört zu den am meisten gespielten Online-Tactik-Shootern der Welt.   
Das Spielszenario beschränkt sich darauf, dass zwei Teams bestehend aus Terroristen und Counter-Terroristen sich auf statisch begrenzten Spielfelder einen Kampf, mittels Schusswaffen liefern. Je nach Spielmodus sind die Ziele die es zu erfüllen gilt verschieden. Im Modus Wettkampf, spielen zwei Teams mit jeweils fünf Spielern gegeneinander und versuchen das gegnerische Team an der Zielerfüllung zu hindern. Das Ziel ist abhängig von dem Spielfeld, auf dem gespielt wird. Für den Modus Wettkampf sind zwei Zielsetzungen vordefiniert. In der Zielsetzung „Defusion“ geht es für das Team der Terroristen darum, eine Bombe auf vordefinierte Plätze der Spielwelt zu platzieren. Das gegnerische Team der Counter-Terroristen hat die Aufgabe diese davon abzuhalten. Ist die Bombe erfolgreich platziert, muss das Team der Counter-Terroristen die Bombe innerhalb von 45 Sekunden entschärfen, um die Runde für sich zu entscheiden. Geschieht dies nicht, gilt die Bombe als detoniert und das Team der Terroristen gewinnt einen Punkt. In der Zielsetzung „Counter-Strike“ muss das Team der Counter-Terroristen mindestens eine von zwei Geiseln aus der Hand der Terroristen befreien, und zu einem vordefinierten Platz in der Spielwelt transportieren. Wird dieses Ziel erfüllt, so erhalten die Counterterroristen einen Punkt. Wird eines der beiden Teams, vor erfüllung eines Ziels mittels der Schusswaffen auf null dezimiert, so bekommt das gegnerische Team einen Punkt.  
Gespielt wird nach dem Spielprinzip „Best of 30“. Das bedeutet, sobald eines der Teams 15 Punkte erreicht hat, gilt dieses Team als Sieger des Spiels. Der Spieler hat die Möglichkeit entweder durch ein Server Matching Verfahren mit zufällig ausgewählten Mitspielern zu spielen, oder mit seinen Freunden gegen ein Gegnerteam zu spielen. Das Spiel Counter-Strike besitzt einen integrierten Sprachchat, welcher es den Spielern eines Teams ermöglicht miteinander zu kommunizieren.  
Jedes Team bekommt vor jedem Rundenbeginn eine Zeitspanne von 15 Sekunden, um eine Taktik mit seinen Mitspielern zu besprechen.   
Da eine sprachliche Formulierung der Strategie der momentanen Runde seitens der Mitspieler in den meisten Fällen nicht ausreicht um eine zum Sieg führende Strategie auszuarbeiten, benötigte es für dieses Problem eine noch nicht vorhandene Lösung. Ein Konzept zur Lösung dieses Problems wird mit dieser Bachelorarbeit ausgearbeitet, und in den nächsten Kapiteln erläutert.

Arbeitsteilung

Wie im letzten Kapitel erläutert, fehlte bissweilen eine Plattform, auf der Spieler des Spiels Counter-Strike Strategien miteinander ausarbeiten können. Um mit der Anwendung möglichst viele Spieler zu erreichen, bedarf es an einem Plattformübergreifende Ansatz. Aus diesem Grund wird diese verteilte Anwendung sowohl als Webseite als auch in Form einer mobilen Applikation umgesetzt.   
Eine Webseite kann von jedem Internetbrowser interpretiert und ausgeführt werden. Bei dieser Art von Bedienung erfordert es keinerlei Installation, und kann Betriebssystem unabhängig genutzt werden.   
Bei der mobilen Applikation wird es sich um eine auf Android basierte Anwendung handeln, welche von jedem androidgestütztem Smartphone benutzt werden kann. Diese Applikation soll es den Anwendern eines mobilen Endgerätes ermöglichen, dieselben Funktionalitäten benutzen zu können, wie die Benutzer der Webseite. Diese Funktionalitäten werden auf die Bildschirmauflösungen und Gerätegrößen angepasst und optimiert sein. Die mobile Applikation erfüllt jedoch anwendungsspezifisch einen anderen Zweck als die Internetseite. Die Internetseite ist in erster Linie dafür gedacht, dass die Spieler Strategien im Vorfeld ausarbeiten und diese dann eventuell miteinander besprechen und optimieren. Da Counter-Strike als Anwendung auf dem PC läuft, ist es umständlich sich in den 15 Sekunden vor beginn eines Spiels eine Strategie im Internetbrowser zurecht zu arbeiten. Hierfür bietet die Mobile Applikation den Vorteil, auf einem unabhängigen Gerät zu laufen. Das Team bekommt somit zusätzlich die Option, während des Spiels mittels der Applikation zu kommunizieren.

Zielsetzung

Grundlagen

Verwendete Programmier und Strukturiersprachen

Javascript

HTML und CSS

Verwendete Frameworks

JQuery

JQuery ist eine von John Resig entwickelte Javascript Bibliothek. Sie wurde 2006 das erste Mal auf einem barcamp in New York vorgestellt. Ein Barcamp beschreibt eine für die Gesellschaft offen gehaltene Tagung in der Informatik. JQuery wurde dafür entwickelt um den Entwicklern Javascript basierter Internetseiten den Zugriff auf die Elemente einer Webseite zu erleichtern. Mittlerweile hat sich John Resig aus dem Projekt zurückgezogen, und die Weiterentwicklung wird von Freiwilligen Mitarbeitern der JQuery Foundation gestützt und umgesetzt. Die JQuery Foundation bezieht den größten Teil ihrer Finanzierung aus Spendenpools privater und kommerzieller Investoren.

Außerdem finanziert sich die Foundation durch die Zusammenarbeit mit Firmen, welche ein JQuery gestütztes System zu kommerzieller Nutzung verwenden. JQuery wurde als Open-Source Software entwickelt, und unterliegt der MIT-License. Als Open-Source Softwareprojekt versteht man, dass der Quellcode der zu entwickelnden Software für die Öffentlichkeit zugänglich gemacht wird. Ein Vorteil dieser Art Software zu vertreiben, besteht darin, das die Anwender dieser Software den Quellcode nachlesen und modifizieren können. Die MIT-License ist eine Lizenzbedingung, welche allen Nutzern dieser Software uneingeschränkt gestattet , diese zu verwenden, zu kopieren, zu verändern, zusammenzufügen, zu veröffentlichen, zu verbreiten, zu unterlizenzieren und/oder zu verkaufen. Die erste Version von JQuery fing mit der Versionsnummer 1.0 an, und ist mittlerweile auf dem Stand 2.2.0. Als im April 2013 die Version 2.0 vorgestellt wurde, lieferte diese eine optimierte verarbeitung der Dato unterstützen Javascript Schnittstelle der Internetbrowser. Außerdem wurde diese Version aufgrund ihrer optimierten Programmierung im Quellcode kleiner, als ihre Vorgänger. Der ausschlaggebende Punkt für die Entwicklung von JQuery 2.0 war die herausgabe der neuen Internetbrowsergenerationen aller namentlich bekannten Entwickler, wie z.B Mozilla Firefox und der Internet Explorer von Microsoft. Da die Versionen 1.0 bis 1.9 bis zu diesem Zeitpunkt von vielen Internetseiten verwendet wurden, entschloss sich die JQuery Foundation, diese weiterhin zu unterstützen und weiterzuentwickeln. Mit der Version 2.0 konnte es auf älteren Browsern zu Einschränkungen in der Funktionsweise von JQuery kommen. Dies war einer der Hauptgründe weshalb die Versionen 1.xx, wie zuvor erläutert, weiter unterstützt wurde.  
  
JQuery greift auf die DOM API zu, welche von jedem Internetbrowser zur verarbeitung HTML gestützter Dokumente bereitgestellt wird. Das DOM (Document Object Model) Ist verantwortlich für die internet Gliederung einer Browserseite. Eine mit HTML strukturiertes Internetseite bildet alle Elemente, welche auf dieser Seite vorkommen intern als Objekte ab. Wird beispielsweise ein Paragraf auf einer Internetseite dargstellt, so befindet sich im DOM ein Element mit dem Tag „<p>“. Man kann sich somit vorstellen, dass ein DOM ein baumstrukturartiges Schema ist, welches beliebig viele Knoten besitzen kann. Dabei kann jedes Element ein übergeordneter Knoten sein, dem weitere Knoten folgen. Jeder dieser Knoten können nun beliebig viele Unterknoten zugeordnet sein. Alle Knoten in einem Browserdokument sind aus dem Grund als Objekt hinterlegt, damit auf diese mittels einer javascript gestützten API zugegriffen werden kann. Jeder javascript fähige Browser besitzt eine javascript Implementierung der DOM API. Diese ermöglich es die Objekte der DOM zu manipulieren. Um diese Elemente manipulieren zu können, wird Javascript benötigt. JQuery baut komplett auf nativen Javascript auf, und bietet dem Entwickler einen komfortableren Zugriff auf die DOM Objekte eines Webdokuments. Mittels JQuery lassen sich alle Attribute eines DOM Objektes auslesen, verarbeiten und modifizieren. Da jeder Internetbrowser unterschiedliche versionen der Javascript Implementierungen der DOM API unterstützen, lassen sich Fehler, welche durch natives Javascript in diesem Zusammenhang auftreten können zum großen Teil mit JQuery abfangen.

JQueryUI

SocketIO

Bootstrap

Datenformat

JSON

Konzeption

Anforderungserhebung

Klassendiagramm

Usecaseanalyse

Design

Designkonzeption

Benutzerführung

Mokups

Spezifikation

Implementierung

Entwurf

Qualitätssicherung

Funktionstests

Usability-Test

Fazit

Ausblick

Literaturverzeichnis

Cameron, D., 2015. *HTML5, JavaScript.* 1. Auflage Hrsg. Heidelberg: dpunkt.verlag GmbH .

Methvin, D., April 18, 2013. *JQuery - Write less, do more..* [Online]   
Available at: https://blog.jquery.com/2013/04/18/jquery-2-0-released/  
[Zugriff am 21 02 2016].