|  |
| --- |
| Lukas Stephan Rieger, Medientechnik Bachelor, 5. Semester, Matrikelnummer: 00741492 |

|  |  |  |
| --- | --- | --- |
|  | Ausarbeitung des ChatHTMLSpotters |  |
|  | |  |
| [20.01.21] | |  |
| Projekt  ChatHTMLSpotter | |  |
| Lastenheft | |  |
|  | |  |
| [Winterfeldt, Kurpis] | |  |
| Authored by: Übersicht | |  |
|  | |  |

Projekt

Contents

[1 Übersicht 2](#_Toc528220584)

[1.1 Projektübersicht 2](#_Toc528220585)

[1.2 Rahmenbedingungen 2](#_Toc528220586)

[1.3 Projektbeteiligte 3](#_Toc528220587)

[2 Funktionelle Anforderungen 3](#_Toc528220588)

[3 Technische Anforderungen 3](#_Toc528220589)

[4 Anforderungen an die Anwendungsschnittstelle 3](#_Toc528220590)

[5 Sonstiges 3](#_Toc528220591)

[5.1 Zusätzliche Anforderungen 3](#_Toc528220592)

[5.2 Organisation 4](#_Toc528220593)

# Übersicht

## Projektübersicht

Auf der Grundarchitektur des vorgelegten Wordspotter soll eine optisch ansprechende und individuelle Webanwendung entstehen. Ich habe es mir zum Ziel gesetzt ein textbasiertes Spiel zu erstellen. Dieses lotst den Anwender auch durch das Labyrinth, in dem es an jeder Kreuzung den Nutzer näheres über jede Richtung, in die man gehen kann, verrät.

## Rahmenbedingungen

Anwender sollen alle textbasierten Spielliebhaber sowie auch Labyrinthdetektive sein. Die Geschichte lässt den Nutzer durch eine Geschichte durch das Labyrinth navigieren.

Es ist für PC- / Laptopnutzer gedacht.

Im Rahmen der Vorlesung Softwareengineering im 5.Semester im Studiengang Medientechnik an der Technischen Hochschule in Deggendorf soll das Projekt als Leistungsnachweis dienen unter Aufsicht von Herrn Götz Winterfeldt.

Das Projekt soll in der Zeit der Praktika der Softwareengineering entstehen und entspricht 2 Semesterwochenstunden Arbeitsaufwand. Es steht kein Budget zur Verfügung.

## Projektbeteiligte

* Lukas Stephan Rieger: [lukas.rieger@stud.th-deg.de](mailto:lukas.rieger@stud.th-deg.de)
* Götz Winterfeldt: [goetz.winterfeldt@th-deg.de](mailto:goetz.winterfeldt@th-deg.de)

# Funktionelle Anforderungen

Das Projekt soll mit der Open-Source-JavaScript-Laufzeitumgebung “node.js” durch einen Terminalaufruf in der Directory von “staticExpress.js” und dem Befehl “node staticExpress.js” auf dem lokalen Rechner einen lokalen Server erstellen, der über localhost:8081 erreichbar ist.

Als IDE wird Visual Studio Code verwendet.

Das Userinterface des Projektes setzt sich aus der “Info.html”,”index.css”und der ”index.js” zusammen. Das Backend besteht aus der

1. “bot.js”, die die abgeschickte Nachricht mit den “Intents” der “test.json” vergleicht und gibt bei Vorkommen die entsprechende “answer” an die “index.html” zurück und bei Nichtvorkommen eine Fehlermeldung an die “index.html” sendet,

2. “staticExpress” , die den lokalen Server aufbaut und auch das Express – Node.js Framework miteinbezieht

3. “test.json” , die wie eine Datenbank funktioniert und auf “intent” / Schlagwörter wie “left way” mit einer “answer” / Antwort reagiert.

Der ChatHTMLSpotter kann aus der eingetragenen und abgeschickten Nachricht die “intents” herausfiltern und gibt dann die entsprechende Antwort zurück. Sollte dieser allerdings nicht vorkommen greift “index.js” ein und überprüft, ob der Anwender nicht gerade durch das Labyrinth navigieren will und wenn das nicht der Fall ist, wird eine Fehlermeldung zurückgegeben. Diese Antwort wird dann in das div mit der ID=”msg” geschrieben und durch CSS formatiert.

Man könnte auch Fragen allgemein über Labyrinthe stellen wie “Was sind Labyrinth?” oder “Wie sind sie entstanden?”. Diese enthalten dann Schlagwörter wie “was” oder “entstanden”, dadurch schaltet sich der bot ein und gibt dann die Antwort wieder, wenn allerdings das abgeschickte Wort nicht länger als 3 Buchstaben ist und es nur aus t,r,f,l, oder a besteht greift “index.js” ein und überprüft wo sich der User gerade im Labyrinth befindet durch “increment” und wenn dann die Kombination der Buchstaben noch dazu passt, wird durch CSS-Veränderungen im Labyrinth weitergegangen. Wenn man auf eine Weggabelung trifft, hat der Nutzer die Wahl sich näheres über den jeweiligen Weg zu holen oder einfach ohne Geschichte weitergeht.

Dazu kommt noch, dass sich das Labyrinth erst durch Erkunden langsam zeigt, dies ist mit CSS-Boxen gelöst, die aufgelöst werden sollte der Nutzer sich einen Block davor oder daneben befinden.

Dieses Programm funktioniert auf allen Betriebssystemen. Entwickelt wurde es teils unter MacOs aber auch auf Windows 10.

Es ist für den PC-Nutzer entwickelt.

# Anforderungen an die Anwendungsschnittstelle

Es muss Node.js und Express.js installiert sein. Es ist keine bestehende Internetverbindung notwendig da es lokal gehostet wird.

Die Anwendung muss hauptsächlich mit der Tastatur bedient werden, aber die Maus wird auch gebraucht.

Die Corporate Identity wird durch einen einfachen Farbverlauf im Hintergrund von hell- auf waldgrün dargestellt. Dies soll die Farbe des Waldes und der Bäume bzw. Der Erde darstellen und soll sich für den User authentischer und einfühlsamer gestalten.

Die Elemente sollen durch die abgerundeten Ecken Einfachheit und auch optisch ansprechend sein. Das linke Element ist das Textausgabefeld und ist durch einen Braunton dargestellt, der an Bäume erinnern soll. Das ganz rechte Element ist dagegen im Kontrast zum Hintergrund auf die entgegengesetzte Hintergrundfarbe eingestellt. In der Mitte ist eine Schriftrolle mit den Anweisungen platziert, die durch ihr Design an lang vergangene Zeiten erinnern soll.

# Sonstiges

## Zusätzliche Anforderungen

Wartungs-, Service- und Hosting-aufwand ist nicht vorhanden, da es sich um einen lokalen Server handelt. Es ist lediglich die oben genannten Bibliotheken zu installieren und alles andere ist lokal gespeichert.

## Organisation

Für das Projekt wurden im Rahmen des 5.Semesters Medientechnik der Technischen Hochschule Deggendorf in der Vorlesung Softwareengineering die Grundlagen gebildet und es wurde gemeinsam die Grundarchitektur erarbeitet.

Projektbeginn war zum Start des 5.Semesters am 01.10.20.

Projektende fällt mit der Abgabefrist zusammen und ist am 21.01.2021

Das Projekt umfasst 2 Semesterwochenstunden und erfolgt nach Selbsteinteilung.