

## **Team 3**

Andreas Glaser

Andreas Khouri

Christa Strauß

Dzianis Brysiuk

Noah Lehmann

Tina Amann

## **Web-Development 2 (Prof. Dr. Jürgen Heym)**

**Sommersemester 2018**

## Inhaltsverzeichnis

1 Einleitung.....	4
2 Beschreibung des Projektes.....	4
Einleitung in die Thematik.....	4
Erklärung des Projektumfangs.....	5
3 Projektrandbedingungen.....	7
Rollenverteilung der Beteiligten.....	7
Vorgehensmodell.....	9
Vorgegebene Hard-/ Software.....	9
Entwicklertools.....	10
4 Erkundung der Frameworks.....	12
HumHub.....	12
Entscheidungsmatrix.....	12
Erklärung der Vorgehensweise.....	12
Bewertungskriterien.....	13
Bewertung.....	14
Entscheidung.....	14
Markdown.....	15
Syntax.....	15
Nutzungsregeln.....	16
AngularJS.....	17
Analyse der Funktionen von AngularJS.....	17
1. Services.....	17
2. Filter.....	19
Bootstrap.....	20
Yii.....	21
5 Entwicklung des Prototypen.....	22
Regelsätze.....	22
Regelsatz 1.....	22
Punktevergabe.....	22
<i>Multiple-Choice</i> .....	23
Frage-Antwort.....	24
Kollaboration.....	26
Punkteverhältnis.....	27
Amateur (Level 1).....	28
Fortgeschritten (Level 2).....	29
Profi (Level 3).....	29
Regelsatz 2.....	31
Warum die KI hilft bzw. Hilferufe simuliert.....	31
Hilferufe.....	31
Wo die KI hilft.....	32
Wann die KI hilft.....	32
Wie die KI hilft beim Kauf von Hinweisen.....	33

Hilfestellung bei einem Rundungsfehler.....	34
Hilfestellung bei einer richtigen Antwort.....	35
Hilfestellung bei keiner richtigen Antwort.....	35
Wie die KI beim Antworten auf Hilferufe hilft.....	35
Wie die KI Hilferufe generiert.....	36
Weitere Beispiele für käufliche Hinweise.....	36
Regelsatz 3.....	38
Beschreibung des Problems.....	38
Was unterscheidet gute von schlechten Spielern?.....	38
Aufbau eines Fragezyklus.....	39
Berechnung von Richtwerten zur Bewertung von Spielern.....	41
Beispiel.....	43
Screendesign.....	44
User Stories.....	44
6 Backup- / Restore-Vorgang.....	45
Erstellen eines Backups.....	45
Wiederherstellen eines Backups.....	46
Backup-Strategie.....	47
7 Termine und zeitliche Abläufe.....	49
Meetings und Teamtreffen.....	49
8 Verweise und Quellen.....	51
9 Anhang.....	52
Sprintplanungen.....	52
Erster Sprint.....	52
Aufgaben.....	52
Rückblick.....	53
10 Erklärung zur Selbstarbeit.....	55

# **1 Einleitung**

Dieser Projektplan beschreibt die grundlegenden Eigenschaften dieses Projektes. Er geht auf die Aufgabe an sich ein, die Absprechen die mit dem Auftraggeber gemacht wurden und beschreibt im Abschluss unsere Herangehensweise an das Projekt. Genaue Daten zu Meilensteinen und Meetings werden nicht hier festgehalten, da das Team sich in Absprache mit dem Auftraggeber im Vornherein auf ein agiles Vorgehen geeinigt hat.

# **2 Beschreibung des Projektes**

In diesem Kapitel wird beschrieben, wie die Randbedingungen des Projektes aufgebaut sind. Der Ursprung und das Ziel des Projektes werden hier erläutert.

## **Einleitung in die Thematik**

Die Grundlegende Idee hinter diesem Projekt liegt in einem Problem, das aktuell die Hochschulen weltweit, aber vor Allem die in den Vereinigten Staaten von Amerika beschäftigt. Viele junge Menschen, die an weiterführende Schulen wie Universitäten gehen, haben große Mühe damit, sich in mathematischen Gebieten ihrer Studienfächer dem Tempo der Lehre anzupassen. Leider fängt dieser Rückstand dieser Lücken oft schon sehr früh an, entweder bei der Einführung der Mathematik an den Grundschulen oder an den weiterführenden Kursen in der High School oder ähnlichen Einrichtungen.

Um dieses Problem anzugehen, haben sich die Penn State University aus Pennsylvania, USA, und die Hochschule für angewandte Wissenschaften aus

Hof, Deutschland, zu einer Zusammenarbeit entschieden. Die führenden Professoren haben sich schnell dazu entschieden, dieses Problem mit der Entwicklung eines Spieles anzugehen. Die Idee ist es, junge Menschen spielerisch dazu zu bewegen, sich in ein Gebiet einzuarbeiten, welches ihnen oft nur wenig Spaß macht. Zusätzlich wird Wert darauf gelegt, dass Schüler sich untereinander helfen und einige Probleme auch zusammen angehen. Im weiteren wird weiter darauf eingegangen, wie die genaue Aufgabe dieses Projektes definiert ist.

## **Erklärung des Projektumfangs**

Wie oben beschrieben, beschäftigt sich dieses Projekt mit der Entwicklung eines Lernprogrammes, welches es Schülern ermöglicht, sich spielerisch und unter Zusammenarbeit in die Thematik der Mathematik einzuarbeiten. Das Projekt hat mit der Absprache einiger Studenten der Hochschule Hof bei den Kollegen aus der Penn State University angefangen. Dort wurden erste Ideen entwickelt, die beschreiben, wie man ein Spiel aufbauen kann, welches mathematischen Inhalt vermittelt. Die Studenten aus der Hochschule in Hof wurden darauf hin beauftragt, sich einer dieser Ideen zu widmen und einen Prototypen zu entwickeln, der es ermöglicht, das Projekt zu legitimieren und die Funktion der Idee des spielerisch-kollaborativen Lernens zu beweisen. Dieses Projekt beschäftigt sich also damit, eine erste Version des Mathe-Spiels zu implementieren. Hier wird jedoch nicht nur auf den spielerischen Inhalt geachtet, sondern der Fokus liegt durchaus zu großen Teilen auf dem kollaborativen Aspekt des Prototyps. Der Prototyp soll umfassende Ideen und Analysen zu dem Konzept des Mathe-Spiels enthalten, die Arbeiten der Studenten sollten umfassend dokumentiert werden, um späteren Teams die Möglichkeit zu geben, auf den Arbeiten des Prototyps aufzubauen. Des weiteren soll am Ende des Projektes ein fertiger und lauffähiger Software Prototyp stehen, der die Kernfunktionen und Ideen des aktuellen Projektteams

widerspiegelt. Details zu den Ideen müssen nicht komplett implementiert werden, dazu ist der Projektzeitraum zu klein.

### 3 Projektrandbedingungen

Es wird das Team vorgestellt und auf die Mittel, die das Team nutzt, eingegangen. Darunter fallen die Rollenverteilung im Projekt, die Vorgehensweise und die genutzten Entwicklertools.

#### Rollenverteilung der Beteiligten

Das Projekt in diesem Umfang ist im Allgemeine auf 7 beteiligte begrenzt. Allen voran ist hier der Auftraggeber Prof. Dr. Jürgen Heym zu erwähnen, er ist verantwortlich für die Idee und Zusammenarbeit mit der Penn State University. Es obliegt ihm, den Auftrag für das Entwicklerteam zu definieren und im Laufe des Projektes erforderliche Änderungen im Umfang der Aufgaben durchzuführen.

Des weiteren wurde von Prof. Dr. Heym ein Teamleiter ernannt, der zuständig für den Ablauf des Projektes ist. Diese Rolle wurde Noah Lehmann zugeteilt, da er der Gruppe angehört hat, die bei der Absprache mit der Penn State University anwesend waren. Seine Aufgabe besteht also primär darin, dem Rest des Teams den Hintergrund des Projektes zu vermitteln und den Ablauf der Arbeit zu beobachten. Um dies zu ermöglichen fungiert er bei Fragen oder Unklarheiten als Vermittler zwischen dem Entwicklerteam und dem Auftraggeber. Zudem wurde ihm im ersten Teammeeting unter Absprache des Entwicklerteams die Aufgabe zugeteilt, die Dokumentation des Prototypen zu verfassen und primär zu verantworten.

Bei dem Teammeeting wurde Dzianis Brysiuk die aus der Vorgehensweise resultierende Aufgabe des Scrummasters übertragen. Seine Aufgabe besteht darin, dem Teamleiter bei der Koordinierung der Aufgaben zu helfen. Zudem ist es seine Aufgabe, nicht nur allgemeine Projektaufgaben zu übernehmen, sondern dem Rest des Teams primär bei ihren Aufgaben zu helfen. Sollten also

irgendwo Unklarheiten entstehen oder Probleme verursacht werden, so steht der Scrummaster dem Team unterstützend zur Verfügung.

Andreas Glaser wird Scrummaster und Teamleiter beratend zur Seite stehen, da er derjenige ist, der im Team die meiste Erfahrung mit dem Vorgehensmodell Scrum ist. Zusätzlich übernimmt er die Aufgabe eines Entwicklers.

Christa Strauß hat die Verantwortung für das Testen der Codeabschnitte übernommen. Zudem arbeitet sie auch als Entwicklerin.

Tina Amann hat die Aufgabe des Entwickelns von Scripten auf dem Server übertragen bekommen, auch sie wird nebenbei als Entwicklerin fungieren.

Andreas Khouri übernimmt das Sichern des Zustands des Web-Servers, auf dem die Applikation laufen wird. Er ist dafür zuständig, einen Backup Plan zu erstellen, außerdem ist es im Verlauf des Projektes seine Aufgabe, die Backups dann zu erstellen und die Speicherabbilder zu sichern. Im Falle eines Problems muss er dann auch die Abbilder wiederherstellen.

### **Zusammenfassende Tabelle der Rollenverteilung:**

<b>Name des Beteiligten</b>	<b>Aufgabenzuteilung</b>
Prof. Dr. Jürgen Heym	Auftraggeber
Andreas Glaser	Entwickler, Beratung bei Fragen zu Scrum
Andreas Khouri	Verantwortlicher für Backups/ Recovery
Christa Strauß	Entwicklerin, Verantwortliche für Tests
Dzianis Brysiuk	Scrum Master
Noah Lehmann	Teamleiter, Verantwortlich für Dokumentation
Tina Amann	Entwicklerin, Verantwortlich für Serverseitige Scripts



## Vorgehensmodell

Nach Absprachen mit dem Auftraggeber hat sich das Projektteam für das agile Vorgehensmodell Scrum entschieden. Dieses ist aus kurzen Sprints aufgebaut, in denen vom Auftraggeber priorisierte Aufgaben abgearbeitet werden. Diese Aufgaben werden regelmäßig mit den Verantwortlichen evaluiert und abschließend neu priorisiert. Dies birgt den Vorteil, dass der Fokus in dem Prototypen immer neu gelegt werden kann. Sollte sich im Laufe eines Sprints herausstellen, dass das Team seine Arbeitszeit auf Dinge fokussiert, die in der Hierarchie nicht ganz oben stehen, so kann dies schnell behoben werden. Die Schere des Verständnisses zwischen Auftraggeber und Team wird so immer gering gehalten. Ein weiterer Vorteil ist, dass das Team bei gescheiterten Ansätzen, was bei einer neuen Entwicklung oft vorkommt, diesen Rückstand schnell verarbeiten kann und die Umstellung auf einen neuen Ansatz schnell und ohne großen Planungsaufwand geschehen kann.

## Vorgegebene Hard-/ Software

Als Grundlage für deren Arbeit wurde dem Entwicklerteam von Auftraggeber Prof. Dr. Jürgen Heym ein Serverzugang auf eine virtuelle Maschine zugeteilt. Diese virtuelle Maschine läuft unter der Virtualisierungssoftware VMWare (von VMWare Inc.). Installiert ist ein Linux Server, der auf der Debian Distribution Ubuntu 16.04.4 basiert. Zugang zu diesem gibt es auf 2 Wege, per Secure Shell im Hochschulnetz einwählen oder über das selbe Netzwerk die Weboberfläche der VMWare zugreifen. Nachinstalliert wurde auf dem Server zusätzlich ein Apache Web-Server der Version 2.4.18 und ein MySQL Datenbank-Server der Version 5.7.20-0. Auf diesem hinterlegt ist eine HumHub Version, die auf dem Server nachinstalliert werden muss. Über das HumHub Modul soll im Projekt der Teil der Kollaboration durchgeführt werden, da dieses

die wichtigsten Funktionen, wie Foren ähnliche Spaces, Umfragen und Aufgaben in Form einer Webbasierten Seite schon zur Verfügung stellt. Die Rechtfertigung der Nutzung von HumHub gegenüber anderer Alternativen ist von den Nachforschungen, die von einem Team studentischer Hilfskräfte vor dem Projektstart durchgeführt wurden, bestätigt.

Zum Projektstart wurden vom Auftraggeber zusätzlich noch weitere Anforderungen an die Entwicklung, spezieller an die Programmierung, gestellt. Nach Absprache mit Prof. Dr. Heym nutzt das Team zur Entwicklung nun die Programmiersprache PHP in der Version 7.0.22 für die Serverseitige Programmierung. Zusätzlich wird serverseitig das PHP-Framework Yii2 verwendet. Clientseitig wird für die Oberflächenprogrammierung das Framework Bootstrap verwendet, die genaue Version dazu wird erst im Verlauf des Projektes festgelegt. Das Content Delivery Management wird über das Framework AngularJS programmiert.

## Entwicklertools

Das Entwicklerteam hat sich nach Absprache im ersten Teammeeting auf folgende Entwicklertools geeinigt; Für die Koordinierung der agilen Vorgehensweise wird die Projektmanagementsoftware Trello genutzt. Diese ist insbesondere deshalb so nützlich und wichtig für das Projekt, da sie eine ständige Erstellung von Aufgaben unterstützt, die unterschiedlich verteilt und priorisiert werden können. Die Teammitglieder werden außerdem benachrichtigt, wenn für sie eine neue Aufgabe erstellt wurde. Als Kollaborative Plattform, zur Kommunikation und Verbindung mit dem Auftraggeber wird die soziale Plattform SeeYou der Hochschule Hof verwendet. Dort wurde für das Team ein Space erstellt, in dem sie untereinander kommunizieren können, sowie wichtige Mitteilungen an die anderen Mitglieder weiterleiten können. Außerdem existiert dort auch ein weiterer Space, der für alle Entwicklerteams des Gesamtprojektes nutzbar ist. Dort können alle Mitglieder Fragen an den

Auftraggeber stellen. Dieser hat zusätzlich die Möglichkeit, neue Anforderungen oder generelle Mitteilungen an die Entwicklerteams weiterzuleiten.

## 4 Erkundung der Frameworks

In diesem Kapitel werden die im Projekt zu nutzenden Frameworks untersucht. Es wird festgehalten, inwieweit die einzelnen Module, die uns in den Frameworks zur Verfügung stehen, uns im Projekt weiterhelfen, um die Idee der Kollaboration im Spiel und die Programmierung des Prototypen umzusetzen.

### HumHub

Wie im vorhergehenden Kapitel beschrieben, beruht die Kollaboration in dem Prototypen dieses Projektes auf der Open-Source Software HumHub. Diese Software bietet dem Team eine soziale Plattform, die bereits über Nutzerbehandlung, Login und andere soziale Komponenten verfügt. Außerdem baut HumHub auf dem PHP-Framework Yii auf, das auch zum Umfang des Projektes gehört und auf das später noch einmal genauer eingegangen wird. Im folgenden wird die Plattform HumHub kurz analysiert und es wird auf die wichtigsten Funktionen eingegangen.

### Entscheidungsmatrix

In diesem Kapitel wird auf die einzelnen Module des Systems HumHub eingegangen. Diese werden analysiert und von den Teammitgliedern bewertet. Es soll außerdem auf die Bewertungsmethoden und die Gewichtungen der einzelnen Punkte eingegangen werden. Zum Schluss wird entschieden, welche Module genutzt werden, diese Entscheidung wird dann noch erläutert.

### *Erklärung der Vorgehensweise*

Es werden für das Projekt relevante Kriterien gesammelt, die dann so objektiv wie möglich nach ihrem Einfluss auf die Entwicklungsarbeit priorisiert werden. Danach werden diese Kriterien mit jedem Baustein abgeglichen und es wird darüber entschieden, wie gut ein Modul dieses Kriterium erfüllt. Alle Entscheidungen über die Priorität und das Erfüllen der Kriterien wurden im Team besprochen.

## **Bewertungskriterien**

Die Priorität eines Kriteriums wird in drei Stufen untergliedert, Hoch (3), Mittel (2) und Niedrig (1). Der Wert, mit dem ein Kriterium bewertet wurde, wird dann mit dem Wert des Einflusses multipliziert.

Der Einfluss eines Kriteriums auf das Projekt wird ebenfalls in drei Stufen unterteilt, Hoch (5), Mittel (3) und Niedrig (1).

Das Produkt aus Priorität und Einfluss bildet den Richtwert eines Kriteriums für ein Modul. Alle Kriterien werden danach aufsummiert, das Modul mit dem höchsten Gesamtwert ist demnach am geeignetsten für die Nutzung im Rahmen von HumHub.

<b>Kriterien</b>	<b>Priorität</b>	<b>Einfluss</b>	<b>Polls</b>	<b>Einfluss</b>	<b>Tasks</b>	<b>Einfluss</b>	<b>Streams</b>
Spaßfaktor	2	5	10	3	6	5	10
Umsetzungsmöglichkeit	3	5	15	1	3	1	3
Komplexität (für Entwickler)	3	5	15	3	9	1	3
Benutzerfreundlichkeit	2	5	10	3	6	5	10
Übersichtlichkeit	2	5	10	3	6	5	10
Arbeitsaufwand (für Entwickler)	1	5	5	3	3	1	1
Lernfaktor	3	1	3	5	9	5	15
Fehleranfälligkeit in	1	3	3	3	3	3	3

Programmierung							
Fehleranfälligkeit bei Nutzung	3	1	3	5	15	5	15
Summe			74		60		70

### ***Bewertung***

<b>Bestes Modul</b>	<b>Mittleres Modul</b>	<b>Schlechtestes Modul</b>
---------------------	------------------------	----------------------------

Farben kennzeichnen das Abschneiden der Module.

### ***Entscheidung***

Obwohl die „Polls“ am besten mit 42 Punkten abgeschnitten haben und die „Tasks“ am schlechtesten, entscheiden wir uns alle Drei (Polls, Tasks und Streams) zu verwenden.


„Polls“ sind am besten für Multiple-Choice-Fragen geeignet, wir möchten aber nicht alle Fragen als Multiple-Choice machen. „Tasks“ sind für Fragen mit Antworten gut geeignet und „Streams“ werden wir für Background-Information verwenden. Somit entsteht ein Abwechslungsreiches Spiel.

## Markdown

In diesem Kapitel wird auf die Syntax eingegangen, die in dem zu nutzenden System HumHub genutzt werden kann, um Texte zu formatieren. Außerdem wird kurz erläutert, welche Elemente genutzt und welche vermieden werden sollen.

### Syntax

Formatierung	Eingegebener Text	Zielform
Fett	<b>**Fettschrift**</b>	<b>Fettschrift</b>
Kursiv	<i>*Kursivschrift*</i>	<i>Kursivschrift</i>
Aufzählungslisten	* Aufzählung 1 (Leerzeichen nach dem Stern nicht Vergessen) * Aufzählung 2	<ul style="list-style-type: none"> <li>• Aufzählung 1</li> <li>• Aufzählung 2</li> </ul>
Nummerierte Listen	1. Schritt 1 2. Schritt 2	<ol style="list-style-type: none"> <li>1. Schritt 1</li> <li>2. Schritt 2</li> </ol>
Verschachtelte Listen	* Erste Ebene * zweite Ebene mit 2 Leerzeichen vor dem Stern/Zahl	<ul style="list-style-type: none"> <li>• Erste Ebene <ul style="list-style-type: none"> <li>◦ Unterebene</li> </ul> </li> </ul>
Überschriften	# Überschriftebene 1 ## Überschriftebene 2 ### Überschriftebene 3 #### ....	<b>Überschriftebene 1</b> <b>Überschriftebene 2</b> <b>Überschriftebene 3</b> ...
Blockzitate	> Blockzitate müssen mit einer Leerzeile beginnen und enden	<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonu.</p>

Inlinecode	Beispiel für `Inlinecode`.	<code>Inlinecode</code>
Codeblocks	<pre> ''' Beispiel für Codeblock ''' </pre>	<code>Codeblock</code>
Bilder	<pre> ![alt] (<a href="http://sampleurl.de/logo.png">http://sampleurl.de/logo.png</a>) </pre>	 <p><i>Abbildung 1: Markdown (Quelle fehlt)</i></p>
Links	<pre> [Anzeigetext für Link] (<a href="http://www.google.de">http://www.google.de</a>) </pre>	<a href="http://www.google.de">Anzeigetext für Link</a>
Bilder mit Link	<pre> [![Alt](imageurl)](link) </pre>	 <p><i>Abbildung 2: Markdown (Quelle fehlt)</i></p> <p>Mit Link</p>
Horizontale Linie	---	<hr/>

## Nutzungsregeln



# AngularJS

## Analyse der Funktionen von AngularJS

Im Folgenden werden nun einige Funktionen und Filter von AngularJS aufgezählt und erläutert, welche wir in unserem Projekt verwenden können. Dies stellt den momentanen Stand dar, kann sich jedoch jederzeit verändern.

### 1. Services

In AngularJS ist ein Service eine Funktion oder ein Objekt, das für eine AngularJS-Anwendung verfügbar und auf diese beschränkt ist.

Service	Analyse
<b>\$exceptionHandler:</b> <i>Behandelt Exceptions, indem er diese an die \$log.error weiterleitet.</i>	Dieser Service ist nützlich, wenn beispielsweise ein timeout oder ein anderer Fehler auftritt. Durch den \$exceptionHandler wird gewährleistet, können wir die Fehler in einen Log auf dem Server schreiben, was uns den Umgang mit diesen erleichtern wird.
<b>\$filter:</b>	Siehe 2. Filter.
<b>\$http:</b> <i>Der \$http Dienst ist ein zentraler AngularJS-Dienst, der die Kommunikation mit HTTP-Servern entweder über das XMLHttpRequest-Objekt des Browsers oder über JSONP ermöglicht.</i>	Da wir speziell in unserem Projekt viel mit Servern arbeiten, benötigen wir diesen Service um zwischen den Servern Kommunikation zu gewährleisten.
<b>\$location:</b> <i>Analysiert URL in der Adressleiste des Browsers und stellt diese für die</i>	Mit Hilfe des \$location können wir in unserem Projekt die unschöne und unübersichtliche in eine kurze, prägnante und verständliche URL umwandeln und

<i>Anwendung zur Verfügung. Mit diesem Dienst können Änderungen an der URL übernommen und angezeigt werden.</i>	festlegen. Dies verhilft zu einer schöneren und ansprechenderen Seite. Außerdem kann man sich so den Link zu der Seite schnell merken und auf sie zugreifen.
<b>\$log:</b> <i>Einfacher Service für die Protokollierung. Schreibt Nachricht sicher in die Konsole des Browsers.</i>	Der \$log-Service ist für unser Projekt sehr interessant, da dieser uns ermöglicht, leichter debuggen zu können.
<b>\$parse:</b> <i>Konvertiert AngularJS-Expressions in eine Funktion.</i>	Der parse-Service kann in uns beim Programmieren einige Arbeit ersparen. Dadurch, dass er automatisch in Funktionen konvertiert, müssen wir das nicht manuell machen und sparen Zeit, welche wir wieder in andere Arbeiten stecken können.
<b>\$q:</b> <i>Führt Funktionen asynchron aus und deren Rückgabewerte können nach der Verarbeitung verwendet werden.</i>	Dieser Service findet eine Verwendung bei zum Beispiel dem Collaborations-Stream. Während der User auf eine Antwort von anderen Spielern wartet, kann er in der Zwischenzeit wieder andere Aufgaben bearbeiten und weiter lernen.
<b>\$timeout:</b> <i>window.setTimeout in AngularJS. Alle Exceptions die bei der Verarbeitung der zu verzögernden Funktion auftreten, werden an den \$exceptionHandler delegiert.</i>	Kann in unserem Projekt für eine höhere Sicherheit der Daten sorgen, indem der User beispielsweise nach einer bestimmten Zeit ausgeloggt wird.

## 2. Filter

In AngularJS ist ein Filter eine Formatierungsmöglichkeit für Daten.

Filter	Analyse
<b>Filter:</b> <i>Wählt eine Teilmenge von Elementen aus einem Array aus und gibt diese als ein neues Array zurück.</i>	<p>Dieser Filter kann genutzt werden, wenn man nur einen Teil von in einem Array gespeicherte Daten braucht und mit diesen weiterarbeiten möchte.</p> <p>Beispielsweise kann das beim Einteilen der Mathematik-Aufgaben nützlich sein.</p>
<b>Number:</b> <i>Formatiert eine Zahl zu einem Text.</i>	<p>Number kann sehr sinnvoll sein, um aus Zahlen einen Text zu formatieren. Der Number-Filter ist speziell in unserem Projekt sinnvoll für zum Beispiel Textaufgaben beziehungsweise für die Frage-Antwort-Aufgaben. Hier wäre es nämlich von Vorteil, wenn man die Aufgaben direkt in Text umwandeln kann, um sie besser und leichter zu verarbeiten.</p>
<b>Date:</b> <i>Formatiert ein Datum in ein gewünschtes String-Format um.</i>	<p>Dieser Filter eignet sich sehr gut um Daten zum Beispiel für die Spielstände der Nutzer zu speichern und dieses als String anstatt als Date weiterzubearbeiten. Strings lassen sich in dem meisten Fällen leichter bearbeiten, als ein Date-Format.</p>
<b>Json:</b> <i>Konvertiert JavaScript-Objekte in JSON-Strings um.</i>	<p>JSON wird benötigt, um mit dem HTTP-Server zu interagieren.</p>
<b>Lowercase &amp; Uppercase:</b> <i>Konvertieren Zeichenfolgen in Klein- bzw. Großbuchstaben.</i>	<p>Diese Filter sind sinnvoll um Zeichenfolgen in eine einheitliche Konvertierung zu wandeln. Dies ist hilfreich, wenn zum Beispiel Strings verglichen werden müssen. Dies wäre bei Frage-Antwort-Aufgaben anzuwenden, um eben die eingegeben</p>

	<p>Antworten mit den korrekten Lösungen zu vergleichen.</p> <p>In unserem Fall haben wir uns für <b>lowercase</b> entschieden, da diese Variante definitiv verbreiteter ist und im Englischen die Kleinschreibung die Norm darstellt. Dadurch muss dann im besten Fall nicht viel umgewandelt werden, um die Eingaben vergleichen zu können.</p>
<p><b>OrderBy:</b></p> <p><i>Gibt sortiertes Array zurück.</i></p>	<p>Der OrderBy-Filter wird gebraucht, um Dinge nach einer bestimmten Reihenfolge zu sortieren. Hierbei wäre es denkbar, diesen Filter unter anderem bei der Auswertung des Votings im Stream einzusetzen. Man würde somit direkt ohne großen Aufwand die Antworten nach den Votes sortiert zurückbekommen.</p>

## Quellenverzeichnis:

Filter: Filter components in ng, <https://docs.angularjs.org/api/ng/filter>,  
Abruf am  
24.04.2018

Services: Service components in ng,  
<https://docs.angularjs.org/api/ng/service>, Abruf am  
24.04.2018

## Bootstrap

**Yii**

## 5 Entwicklung des Prototypen

Im folgenden Kapitel wird die Entwicklung des Prototypen dokumentiert. Es werden alle Überlegungen, die in die Spielidee und in das Konzept der Kollaboration einfließen, aufgenommen. Dazu gehören unter Anderem die Regelsätze für die Nutzung der künstlichen Intelligenz und der Vergabe von Punkten, um im Spiel voran zu kommen, Designs der Spiele- und Kollaborationsseiten und User Stories, die die Interaktion mit dem Prototypen verdeutlichen.

### Regelsätze

Hier wird auf die vom Auftraggeber Prof. Dr. Jürgen Heym definierten Regelsätze für das kollaborative Spiel eingegangen. Es werden Alle 3 Regelsätze nochmals formuliert und anschließend mögliche Lösungen präsentiert.

#### Regelsatz 1

Verteilung von Wissenspunkten und Kollaborationspunkten und dessen Verhältnis.

#### ***Punktevergabe***

In dem Spiel werden zwei Arten von Punkten vergeben, zum einen Kollaborationspunkte und zum anderen Wissenspunkte. Kollaborationspunkte erreicht der Nutzer, indem er Fragen beantwortet, die möglicherweise von anderen Spielern gestellt wurden und ihnen beim Lösen von Aufgaben helfen. Wissenspunkte sammelt der User durch das richtige Beantworten von Matheaufgaben, welche diesem im normalen Spielablauf gestellt werden. Wie der Name schon sagt, verdeutlichen Wissenspunkte, dass man sich nach deren Erreichen erfolgreich Wissen angeeignet hat.

Nachkommend werden die einzelnen Arten an Punkte zu gelangen aufgezählt und näher erklärt.

### ***Multiple-Choice***

Multiple-Choice-Fragen sind Fragen, bei denen insgesamt vier Antwortmöglichkeiten vorgegeben sind und eine gewisse Anzahl von diesen zur Lösung der gestellten Aufgabe ausgewählt werden müssen. Dabei ist es vorgegeben, wie viele Antworten benötigt werden. Der Fall von vier richtigen Antwortmöglichkeiten wird jedoch ausgeschlossen, da dies sonst zu keinen Lernerfolg führen würde.

Die Punktevergabe setzt wie folgt zusammen:

Hat der Spieler keine einzige richtige Antwort, bekommt er keinen Punkt.

Gibt es eine einzige richtige Antwort und er wählt diese aus, erhält er drei Wissenspunkte, da die Wahrscheinlichkeit, dass er genau diese anklickt gleich  $\frac{1}{4}$  ist und somit relativ gering. Ist die gewählte Antwort halb richtig, erhält er nur einen Punkt.

Dies wird mithilfe eines Beispiels erklärt. (siehe Abb.1)

Frage: Wie lauten die Nullstellen der folgenden Gleichung?

$$y = (35 - x_1)(x_2 + (-2))$$

- ☐ 35, 2
- ☒ 35, (-2)
- ☐ 7, 3
- ☐ 88, -15

*Abbildung 3: Bsp. halb richtige Antwort (Beispiel von Christa Strauß und Tina Amann)*

Die folgende Antwort ist zur Hälfte richtig, da die erste Nullstelle 35 ist, die zweite aber 2 und nicht -2. Deshalb erhält der User einen Punkt.

Sind zwei Antworten richtig und der Nutzer sucht diese aus, sammelt er zwei Punkte, da eine 50/50 Chance besteht. Entscheidet er sich jedoch nur für eine der Richtigen, kriegt er einen Wissenspunkt.

Zuletzt besteht die Möglichkeit von drei richtigen Antworten. Wählt der Spieler alle drei Richtigen, bekommt er einen Punkt, da in diesem Fall zwei seiner Antworten immer richtig sind und die Wahrscheinlichkeit richtig zu liegen bei  $\frac{3}{4}$  liegt, was sehr hoch ist. Der Nutzer könnte hierbei somit raten und würde dennoch mindestens zwei richtige Antworten ankreuzen. Der Lerneffekt ist dabei sehr gering, weshalb er hier auch nur einen Wissenspunkt verdient. Jedoch wird die Möglichkeit von drei Richtigen nicht ausgeschlossen, da sich der Spieler allein durch das Lesen der Frage und der Antworten schon ein wenig Wissen aneignet, welches er durch weiteres Beantworten von Fragen vertiefen und festigen wird. Ist bei diesem Fall eine Antwort des Nutzers falsch, bekommt er keinen Punkt, da wie oben schon beschrieben, zwei Antworten automatisch immer korrekt sind.

## **Frage-Antwort**

Bei Frage-Antwort-Aufgaben wird dem User eine Frage gestellt, welche er mit eigenen Worten beantworten muss. Dabei kann er in ein vorgegebenes



Textfeld seine Lösung eingeben, welche dann ausgewertet wird. Bei dieser Art von Fragen sammelt der Spieler erneut Wissenspunkte. Diese Art der Aufgaben sind für den User komplexer zu lösen, wie schon in der Entscheidungsmatrix festgehalten, weshalb ihn hier mehr Punkte erwarten als bei Multiple-Choice-Fragen.

Frage: Welchen Wert hat  $x$ ?  
 $x + 10 = 20$

Nutzer-Eingabe: 10

Abbildung 4: Bsp. leichte Aufgabe

Frage: Was ist die Lösungsmenge?  
 $x + 7a = 19a - 25 \quad (a, x \in \mathbb{Q})$

Nutzer-Eingabe:  $L = \{12a - 25\}$

Abbildung 5: Bsp. schwere Aufgabe

Die Frage-Antwort-Aufgaben werden in zwei Kategorien unterteilt, zum einen gibt es schwere (siehe Abb.2) beziehungsweise umfangreichere Aufgaben, zum anderen leichtere und nicht so komplex (siehe Abb.3) zu beantwortende Fragen. Außerdem gibt es Aufgaben mit mehreren möglichen Antworten, welche extra zu behandeln sind. Es wird bei der Aufgabe angemerkt, ob sie eine der schweren oder der Leichten ist, damit der Nutzer weiß, wie viele Punkte ihn erwarten.

Wie immer gilt, gibt der Benutzer eine falsche oder ungültige Antwort, werden ihm keine Punkte gut geschrieben.

Beantwortet der Spieler eine der komplizierteren Fragen korrekt, so erhält er fünf Punkte.

Löst er eines der simpleren Probleme richtig, so sammelt er vier Wissenspunkte.

Bei den Aufgaben mit mehreren Antwortmöglichkeiten werden zwei Arten zusammengefasst. Die Erste beinhaltet das Aufzählen mehrerer Antworten auf eine Frage, die Andere das Darstellen eines mehrschrittigen Rechenweges. Bei beiden gibt es pro richtige Antwort beziehungsweise Rechenschritt vier Punkte. Dies wird darauf begründet, dass die Wahrscheinlichkeit, dass man die Antwort wie bei Multiple-Choice erraten kann, ohne sie zu wissen, gleich null

ist. Das Wissen des Spielers wird bei dieser Fragen-Art wirklich auf den Prüfstand gestellt, da er diese Vorkenntnisse und ohne Hilfe nicht beantworten kann. Trifft er keine korrekte Lösung, bekommt er keinen Punkt. Dies wird nun an einem Beispiel verdeutlicht. (siehe Abb.)

Frage: Was ist die Lösungsmenge?  
 Gib auch Lösungsweg mit an.  
 $x+7a = 19a-25 \quad (a,x \in \mathbb{Q})$

Nutzer-Eingabe:  $x+7a = 19a-25 \quad / -7a$   
 $x = 12a-25$   
 $L = \{12a - 25\}$

*Abbildung 6: Bsp. Fragen-Antwort-Aufgabe mit Rechenweg*

Im oberen Beispiel wird der Lösungsweg verlangt. Das bedeutet die wichtigsten Schritte werden gewertet. Der Nutzer bekommt somit pro Rechenschritt vier Punkte, in diesem Beispiel wären dies acht Punkte.

## **Kollaboration**

Bei einer Kollaboration sammelt man, wie der Name schon sagt, die sogenannten Kollaborationspunkte. Unter einer Kollaboration in diesem Spiel versteht man, die gegenseitige Unterstützung und Zusammenarbeit der Nutzer. Hat ein Spieler beispielsweise ein Problem seine Aufgabe zu lösen oder generell eine Frage zu dem Themengebiet, kann er diese in einem für die Nutzer regenerierten Chat stellen. Auf diese Unterhaltung kann jeder Spieler des selben Levels zugreifen. Anderen Spielern wird die Frage dann angezeigt und können diese, wenn sie das nötige Wissen dazu haben, im Chat

beantworten. Beim Beantworten der Fragen anderer Nutzer, können Spieler ihre Kollaborationspunkte sammeln.

Auf eine Frage ist es mehreren Spielern möglich zu antworten. Für die Antworten können dann ebenfalls die Nutzer voten und die drei Personen mit den meisten Upvotes, also mit den besten Erklärungen, erhalten ihre Punkte. Das Voting nach dem Beantworten der Probleme durch die anderen Nutzer dauert stetig zehn Minuten, danach werden die Punkte verteilt. Antworten weniger als drei Personen oder sogar keine, werden diese Lösungen „bepunktet“ oder die künstliche Intelligenz schreitet ein und liefert eine Hilfestellung auf die Frage.

Die Lösung mit den meisten Upvotes erhält fünf Kollaborationspunkte, da diese Art der Beantwortung ungefähr mit Frage-Antwort-Aufgaben gleichzusetzen ist. Der User muss beides mal eine eigene Antwort beziehungsweise Hilfe formulieren und die Frage sollte jedes Mal korrekt beantwortet werden. Der einzige Unterschied ist, dass der Nutzer Kollaborationspunkte anstatt der Wissenspunkte erhält.

Der zweite Platz bekommt vier Punkte und der dritte drei, was sich eben auf die absteigende Platzierungen zurückführen lässt.

### ***Punkteverhältnis***

Das Basiskonzept des Spiels basiert auf drei Schwierigkeitsstufen: Amateur, Fortgeschritten und Profi.

Die erste Stufe und somit der Einstieg in das jeweils gewählte Themengebiet ist als leichtes Anfängerlevel angedacht. In diesem werden die Nutzer hauptsächlich damit beschäftigt das Thema kennen zu lernen und dieses grob zu verstehen, damit sie in der Lage sind leichte Aufgaben in diesem Bereich zu lösen.

Das darauffolgende Level zielt auf die fortgeschritteneren Kenntnisse ab und vertieft das Wissen des ersten Levels. Hierbei wird auf die jeweils bereits erreichten Grundkenntnisse aufgebaut. Die Aufgaben werden zunehmend schwerer und fordern den User somit immer mehr.

Hat der Spieler dann das erwartete Niveau erreicht, kommt er in eine noch höhere Anforderungsklasse, welche für die Profis auf dem Gebiet gedacht ist. Hier werden besonders anspruchsvolle Aufgaben gestellt, wodurch die erworbenen Kenntnisse des Spielers auf den Prüfstand gestellt und bis ins kleinste Detail verbessert werden. Schließt der Benutzer diese Stufe ab, hat er sehr gute Kenntnisse in dem Themenbereich und kann sich nun anderen Problematiken der Algebra zuwenden.

Im Folgenden werden nun die Verhältnisse der Kollaborations- und Wissenspunkte der jeweiligen Stufen erläutert und begründet.

### ***Amateur (Level 1)***

Das Punkteverhältnis in der Anfängerstufe beträgt 25:75. Dies bedeutet, dass der Nutzer 25 Prozent seiner Punkte durch Kollaboration und 75 Prozent durch Wissen sammeln muss, damit er dieses Level abschließen kann.

Die Auswahl dieses Verhältnisses beruht darauf, dass in dieser Stufe zunächst der Fokus darauf gelegt wird, sich Wissen und Grundkenntnisse anzueignen. Deswegen sollen deutlich mehr Wissenspunkte als Kollaborationspunkte erreicht werden. Ein weiterer Grund ist, dass die Aufgaben dieses Levels noch relativ leicht und in der Norm schnell zu lösen sind, weshalb man in kurzer Zeit im Vergleich mit einem höheren eher Wissenspunkte sammeln kann. Es ist außerdem viel schwerer in dieser Niveaustufe Kollaborationspunkte zu sammeln, da die Aufgaben für viele Nutzer ohne Hilfe zu bewältigen sind und somit die Kollaborationsfunktion nicht sehr oft in Gebrauch genommen wird.

Jedoch wird trotzdem noch Wert auf diese gelegt, da es immer Personen gibt, die sich selbst bei simpleren Problemen schwer tun und Hilfe benötigen.

### ***Fortgeschritten (Level 2)***

Folgendem Level wurde das Verhältnis 50:50 zugeordnet.

In dieser Stufe gleichen sich die Kollaborations- und Wissenspunkte aus, da hier beide Aspekte gleich wichtig sind und benötigt werden. Es muss sowohl das Wissen, das zuvor angeeignet wurde, vertieft als auch Neues erlernt werden.

Die Nutzer, die sich mit den nun mittelschweren Aufgaben etwas schwer tun, haben die Möglichkeit sich durch die Zusammenarbeit mit anderen helfen zu lassen, wodurch die guten Spieler wiederum Kollaborationspunkte sammeln können. Dennoch reißt die Motivation sich neues Wissen anzueignen nicht ab, da Wissenspunkte erreicht werden müssen. Das Level ist somit ausgewogen und sowohl die guten als auch die schlechten Spieler werden gefördert und können dadurch gemeinsam das Ziel der Fortgeschrittenen-Stufe erreichen.

### ***Profi (Level 3)***

In der Profi-Stufe besteht zwischen den Kollaborations- und Wissenspunkten das Verhältnis 65:35.

In dem höchsten Level werden die Fragen auf ein hohes Schwierigkeitsniveau gehoben, somit werden deutlich mehr Nutzer Probleme haben, die Fragen und Aufgaben zu lösen. Durch gegenseitiges Helfen wird das bereits erworbene Wissen angewendet und vor allem vertieft, was besonders das Ziel in dieser Stufe ist. Deshalb ist vor allem die Kollaborationsfunktion, bei der sich die Nutzer gegenseitig unterstützen und helfen können, in diesem Level von hoher

Notwendigkeit, was die hohe Gewichtung der Kollaborationspunkte erklärt. Die Wissenspunkte fallen somit etwas in den Hintergrund, dürfen aber nicht komplett vergessen werden, da der Nutzer stets Motivation benötigt, um weiterhin zu lernen und sich Wissen anzueignen.

## **Regelsatz 2**

Wann greift eine künstliche Intelligenz bei der Kollaboration ein und wie das realisiert werden kann.

### ***Warum die KI hilft bzw. Hilferufe simuliert***

Das Lernsystem beruht auf der Kollaboration der Nutzer, sind allerdings keine Nutzer in der Lage einem hilfebedürftigen Nutzer zu helfen, muss das System diese Aufgabe übernehmen. Hierfür wird eine KI entwickelt, die auf die Hilferufe der Nutzer reagieren und passend antworten kann.

Weiterhin sollen die Nutzer in der Lage sein Kollaborationspunkte zu sammeln, auch wenn kein anderer Nutzer Hilfe benötigt oder die Probleme der anderen Nutzer zu anspruchsvoll sind. Deshalb generiert die KI automatisch Hilfeaufrufe an Spieler, falls nicht genug natürliche Hilferufe gestellt werden, und gibt wiederum Antworten auf die eigenen Fragen, falls keine Antwort von mindestens einem Spieler erfolgt ist. Diese Antworten werden ebenfalls bewertet, um die KI-Handlungen nicht von spielergesteuerten Handlungen unterscheidbar zu machen. Dies hat auch zur Folge, dass das System belebt wirkt und die Nutzer leichter motiviert sind selbst Hilferufe zu generieren und auf die anderer zu antworten.

### ***Hilferufe***

Hilferufe sind ein separater Chat, der von einem Nutzer oder der KI generiert werden kann. Je nachdem wie lange der Hilferuf bereits besteht, können verschiedene Gruppen in dem Chat antworten, näheres hierzu in Regelsatz 3. Jeder Spieler, der auf einen Hilferuf reagiert hat, bekommt 50 Sekunden

nachdem der Hilferuf generiert worden ist, alle abgegebenen Hilfestellungen für fünf Minuten angezeigt, sollte die Gruppe der Nutzer nicht in einem gemeinsamen Zeitraum arbeiten, sondern über den Verlauf einer Woche verteilt, bietet sich hierfür an diese Zeit auf einen Tag zu erhöhen. Die Hilferufe und die dazugehörigen Antworten werden nach einer gewissen Zeit gelöscht, um zu verhindern dass die Nutzer lediglich alte Hilferufe durchforsten anstatt selber Hilferufe zu generieren

### ***Wo die KI hilft***

Die KI hilft, wenn ein Nutzer einen Hilferuf generiert hat und kein anderer Nutzer ihm helfen konnte. Außerdem stellt die KI generierte Hinweise zu allen Aufgaben zum Kauf zur Verfügung.

### ***Wann die KI hilft***

Es wird kurz dargestellt, wann die künstliche Intelligenz eingreift und weshalb.

#### **Beim Kauf eines Hinweises**

Um dem Nutzer die Antwort nicht vorwegzunehmen, soll die KI erst einschreiten, wenn dieser um Hilfe bittet. Dies teilt der Nutzer mit, indem er auf den „Hilfe“-Button klickt, er wird hieraufhin gefragt ob er einen Hinweis kaufen möchte.

#### **Nach dem Generieren eines Hilferufs**

Damit die anderen Nutzer eine Möglichkeit haben, Hilfestellung zu leisten und so erfolgreich Kollaboration zu betreiben, soll die KI 50 Sekunden warten bevor sie dem hilfebedürftigen Nutzer hilft. Diese Zeit ist bewusst lang gewählt um auch Nutzer, die das Zehnfingersystem noch nicht beherrschen, nicht zu



benachteiligen. Sie darf aber auch nicht zu lange ausfallen, da der hilfebedürftige Nutzer sonst versucht ist, sich mit neuen Problemen zu beschäftigen.

### ***Wie die KI hilft beim Kauf von Hinweisen***

Die KI soll dem Nutzer anfangs nur Hinweise in die richtige Richtung geben, soll aber bei Bedarf den Nutzer durch die Aufgabe leiten können. Deshalb sind die Hinweise in kleine Teile zerlegt und müssen vom Nutzer einzeln gekauft werden, hierdurch können schlechtere Nutzer durch die Aufgabe geleitet werden und bessere müssen nur die Hinweise kaufen, die sie auch benötigen.

### **Wenn der Nutzer keine Lösung hat**

Hat der Nutzer keinen Lösungsansatz finden können, kann ihm die KI eine Reihe von, für diese Aufgabe generierten, Hinweisen geben.

Beispiel:

Aufgabe:

$$0 = -20x^2 + 320$$

kleineres  $x =$  ; größeres  $x =$

Hinweise:

1. Versuche von beiden Seiten der Gleichung 320 zu subtrahieren.
2. Versuche beide Seiten der Gleichung durch 20 zu dividieren.
3. Versuche beide Seiten der Gleichung durch -1 zu dividieren.

4. Versuche von beiden Seiten der Gleichung die Wurzel zu ziehen.

1. Möglichkeit: Die Hinweise müssen in der richtigen Reihenfolge kommen und vom Nutzer mit den Wissenspunkten, die er für die Aufgabe erhalten würde, bezahlt werden. Ein Hinweis soll einen Wissenspunkt kosten, der Nutzer kann diese Punkte zurück erhalten, indem er die Aufgabe, eine Woche später, korrekt löst.

2. Möglichkeit: Die Hinweise müssen in der richtigen Reihenfolge kommen und vom Nutzer mit Kollaborationspunkten bezahlt werden. Ein Hinweis soll einen Kollaborationspunkt kosten, der Nutzer kann diese Punkte zurück erhalten, indem er auf Hilferufe anderer Nutzer oder der KI reagiert.

### **Wenn der Nutzer eine falsche Lösung hat**

Ist eine Antwort, auf die Aufgabe des Nutzers vorhanden, so kann die KI eine seinem Fortschritt angepasste Hilfestellung bieten.

Aufgabe:

$$0 = -10x^2 + 320$$

Ergebnis auf zwei Nachkommastellen gerundet.

kleineres  $x =$  ; größeres  $x =$

### ***Hilfestellung bei einem Rundungsfehler***

Antwort des Nutzers: kleineres  $x = -5,65$  ; größeres  $x = 5,65$

Hilfestellung: Ab 5 wird aufgerundet, nur 4 oder kleiner wird abgerundet.

### ***Hilfestellung bei einer richtigen Antwort***

Antwort des Nutzers: kleineres  $x = -2,66$  ; größeres  $x = 5,66$

Hilfestellung: Das kleinere  $x$  ist nicht korrekt, das größere hingegen ist korrekt. (Diese Hilfestellung ist für den Nutzer kostenlos, bei Bedarf kann er die selben Hinweise kaufen, die er bekommen würde, wenn er keine Antwort gegeben hätte.)

### ***Hilfestellung bei keiner richtigen Antwort***

Antwort des Nutzers: kleineres  $x = -2,66$  ; größeres  $x = 2,66$

Hilfestellung: Beide  $x$  sind nicht korrekt. (Diese Hilfestellung ist für den Nutzer kostenlos, bei Bedarf kann er die selben Hinweise kaufen, die er bekommen würde, wenn er keine Antwort gegeben hätte.)

### ***Wie die KI beim Antworten auf Hilferufe hilft***

Um Hilfe leisten zu können, muss die KI zunächst verstehen, wo das Problem des Nutzers liegt. Dies kann zum einen durch Machine Learning aber auch durch das Ausfiltern von Schlüsselwörtern realisiert werden. Machine Learning kann verwendet werden um die Antworten der Nutzer, die auf Hilferufe getätigt wurden zu analysieren, die KI soll lernen, was den Nutzern bei einem bestimmten Problem geholfen hatte und daraus eine Antwort generieren. Das Ausfiltern von Schlüsselwörtern soll ermöglichen, auf passende Kapitel des Lernstoffes zu verweisen, sodass der Nutzer durch Nachlesen des Stoffes das Problem selbständig lösen kann.

## **Wie die KI Hilferufe generiert**

Die KI kann alte archivierte, vorgefertigte oder automatische Hilferufe generieren. Die archivierten Hilferufe sind alte Hilferufe von Nutzern die von Nutzern beantwortet werden konnten, die vorgefertigten sind von einer betreuenden Lehrkraft oder einem Administrator zu erstellen. Automatische Hilferufe müssen von der KI so generiert werden, dass sie dem Niveau der Nutzer entsprechen. Sie bewertet die Antworten der Nutzer auf Basis ihrer Qualität. Wenn Nutzer wenig Kollaborationspunkte haben, generiert die KI für diesen Nutzer gezielt Hilferufe.

## **Weitere Beispiele für käufliche Hinweise**

Im folgenden werden einige Beispiele gezeigt, wie die künstliche Intelligenz den Anwendern bei Bedarf Hinweise geben könnte.

### **Beispiel 1**

Aufgabe:

$$27c = 22c + 5x^2$$

kleineres  $x =$  ; größeres  $x =$

Hinweise:

1. Versuche alle Terme mit  $x$  und alle Terme ohne  $x$  auf verschiedene Seiten zu bringen.
2. Versuche auf beiden Seiten der Gleichung  $22c$  zu subtrahieren.
3. Versuche beide Seiten der Gleichung durch  $5$  zu teilen.
4. Versuche auf beiden Seiten der Gleichung die Wurzel zu ziehen.

## Beispiel 2

Aufgabe:

$$-x + 4a + 14 = 16a$$

x =

Hinweise:

1. Versuche alle Terme mit x und alle Terme ohne x auf verschiedene Seiten zu bringen.
2. Versuche auf beiden Seiten der Gleichung 4a zu subtrahieren.
3. Versuche auf beiden Seiten der Gleichung 14 zu subtrahieren.
4. Versuche beide Seiten der Gleichung durch -1 zu teilen.

## Beispiel 3

Aufgabe:

$$3y + 9a - 6c = 27b$$

y =

Hinweise:

1. Versuche alle Terme mit y und alle Terme ohne y auf verschiedene Seiten zu bringen.
2. Versuche auf beiden Seiten der Gleichung 9a zu subtrahieren.
3. Versuche auf beiden Seiten der Gleichung 6c zu addieren.
4. Versuche beide Seiten der Gleichung durch 3 zu teilen.

## **Regelsatz 3**

Wie wird sichergestellt, dass alle Spieler, unabhängig von Ihren Fortschritten oder Kenntnissen in der Materie, gleichberechtigt Kollaborationspunkte sammeln können.

### ***Beschreibung des Problems***

Das Problem bei kollaborativem Lernen ist, dass das Niveau, auf dem sich die Schüler befinden, stets unterschiedlich ist. Es wird also immer der Fall sein, dass es einem Anwender leichter fällt, im Spiel voran zu kommen, als es einem schwächeren Schüler fällt. Das Grundkonzept des Spieles ist es ja, laut Vorgabe des Auftraggebers Prof. Dr. Heym, dass Spieler sowohl durch Fortschritte im Inhalt der Mathematik, aber auch durch Mithilfe für andere Schüler bei Problemen Punkte bekommen, wodurch sie im Spiel voranschreiten (Dieses Konzept wird in Regelsatz 1 beschrieben). Leider fällt es den schwächeren Schülern deutlich schwerer, anderen Mitspielern zu helfen, denn entweder haben sie selbst noch Probleme mit der Materie oder ein besserer Schüler hat ihnen die Möglichkeit der Mithilfe durch schnelleres Antworten vorweggenommen. Der folgende Abschnitt beschäftigt sich genau mit diesem Problem und erörtert mögliche Herangehensweisen, um das gleichmäßige Voranschreiten aller Schüler zu gewährleisten.

### ***Was unterscheidet gute von schlechten Spielern?***

Um zu differenzieren, ob die aktuelle Lage im Spiel ausgewogen und fair ist, muss man sich erst Gedanken darüber machen, was einen guten Spieler von

einem schlechten Spieler unterscheidet. Im allgemeinen soll hier von einem schlechten Schüler gesprochen werden, wenn dieser weniger als 60% der bisher erreichbaren Wissenpunkte im Spiel erlangt hat. Ein gute Spieler hat demnach mehr als 60%.

Wissenspunkte werden im Laufe der Spiele unterschiedlich vergeben, man unterscheidet, ob ein Spieler eine mathematische Aufgabe komplett selbstständig gelöst hat, er einen Tipp oder Mithilfe eines anderen Schülers in Anspruch nehmen musste oder ob er die Aufgabe gar nicht beantworten konnte.

### ***Aufbau eines Fragezyklus***

Der Prozess einer Frage im kollaborativen Umfeld beginnt damit, dass ein Spieler mit seiner Aufgabe nicht voran kommt, er benötigt demnach Hilfe und stellt seine Frage öffentlich an alle anderen Mitspieler. Um das oben genannte Problem zu vermeiden wird der weitere Ablauf zeitlich unterteilt:

Sobald die Frage gestellt wurde, wird ein Timer gestartet. Dieser Timer erlaubt es in den ersten 20 Sekunden der Fragestellung nur den schlechteren Schülern, die Frage zu beantworten, damit sie etwas Vorlaufzeit gegenüber den guten Spielern haben, um eine geeignete Antwort zu verfassen. Sollte in diesen 20 Sekunden keiner die Frage beantwortet haben, so wird die Frage für alle Spieler freigegeben, außer für Spieler, die in der Vergangenheit zu oft und zu schnell auf Fragen geantwortet haben, um allen anderen die Möglichkeit zu nehmen, Kollaborationspunkte zu sammeln. Jetzt haben alle 15 Sekunden Zeit, die Frage zu beantworten. Sollten im Vorfeld Spieler aufgefallen sein, die zu oft und zu schnell Fragen beantworten, dann bekommen diese im Anschluss an die vergangenen 35 Sekunden nochmals 15 Sekunden Zeit, um die Frage zu lösen. Nach diesen 50 Sekunden wird die Frage an die künstliche Intelligenz weitergeleitet, mit der sich Regesatz 2 beschäftigt.

Es sollte noch mit angefügt werden, dass die Frage ab dem Zeitpunkt, in dem sie gestellt wurde, für alle Spieler sichtbar ist, aber immer nur für die oben genannten Zielgruppen beantwortbar ist. Außerdem wird der Timer zum stehen gebracht, sobald ein Spieler anfängt, eine Antwort einzutippen. Somit soll die Unterbrechung durch die künstliche Intelligenz unterbunden werden.

Zum Schluss werden dem Fragestellenden Spieler die Antworten angezeigt. Er hat dann die Möglichkeit, über 3 Icons auszuwählen, wie sehr im die Antwort des anderen Spieler geholfen hat. Es werden drei Smileys angezeigt, ein grüner, der signalisiert, das die Antwort sehr hilfreich war, ein gelber, der klarstellt, das die Antwort durchaus weiter geholfen hat, aber dennoch zu Wünschen übrig lies, und einen roten Smiley, der klarstellt, dass die Antwort der Frage nicht zu genüge getan hat.



*Abbildung 7: Smileys zur Bewertung der Hilfe*

Die Kollaborationspunktevergabe ist in Regelsatz 1 klar beschrieben. Aufbauend auf diesen Werten bekommt ein Spieler für eine hilfreiche Antwort 100% der möglichen Antworten, eine weniger hilfreiche Antwort bekommt aber nur 50% ihrer möglichen Punkte. Eine schlechte Antwort soll aber nicht ganz leer ausgehen, denn der Aufwand, zu antworten soll trotzdem beachtet werden. Deshalb bekommt eine schlechte Antwort 10% ihrer Punkte. Ausnutzen dieser 10% durch sinnfreie Antworten ist allerdings nicht möglich, da dies durch die Bewertung der Spieler, die unnötig antworten, abgefangen wird.



## ***Berechnung von Richtwerten zur Bewertung von Spielern***

Zur Berechnung der Verhältnisse zwischen Antworten bei Spielern werden nur abgeschlossene Kapitel genutzt. In Regelsatz 1 wird die Voraussetzung an Kollaborationspunkten für ein Voranschreiten in das nächste Kapitel beschrieben. Somit wird ein Minimum an Punkten festgehalten. Das Maximum orientiert sich an dem Spieler, der in dem Kapitel die meisten Kollaborationspunkte gesammelt hat und der das Kapitel beendet hat. Aus den Beiden Extremwerten wird nun der Durchschnitt errechnet. Wenn ein Spieler mit seinen Kollaborationspunkten in abgeschlossenen Kapiteln 25% über dem Durchschnitt liegt, so ist dieser Spieler ein Kandidat für solche, die potentiell Fragen zu oft beantworten, um anderen Spielern die Möglichkeit zur Hilfe zu nehmen. Um die Spieler, die den anderen das Kollaborieren erschweren, eindeutig zu identifizieren, werden in den oben errechneten Wert zwei weitere mit eingenommen.

Der erste der beiden Werte setzt sich zusammen aus folgenden Einflüssen:

- Anzahl der Fragen, auf die von aktiven Spielern, nicht von der künstlichen Intelligenz, geantwortet wurde.
- Anzahl der Fragen, die der zu betrachtende Spieler selbst beantwortet hat.
- Gesamtanzahl der teilnehmenden Spieler im System.

Der zweite der beiden zusätzlichen Werte betrachtet die Antwortzeit, die der verdächtige Spieler gebraucht hat, um seine Antworten im Durchschnitt zu geben.

Beispiel Formel für das Errechnen des Wertes, der einen Spieler als übermäßigen Beantworter identifiziert:

- $a$  = Anzahl der von Spielern beantworteten Fragen  
 $b$  = Anzahl der vom zu betrachtenden Spieler beantworteten Fragen  
 $c$  = Anzahl der Spieler, die am System teilhaben

$t$  = Zeit, die der zu betrachtende Spieler im Schnitt für eine Antwort braucht (in s)

wenn  $\frac{a}{c} \leq b * \sqrt{\frac{15 s}{t}}$ , dann ist der Spieler zu eifrig.

## Beispiel

### Aufgabe:

Schreibe die Gleichung, die du erhältst, wenn du in

$$2x + a = b, \text{ mit } (a, b, x \in \mathbb{Q})$$

2 statt a und 10 statt b setzt und bestimme deren Lösungsmenge.

### Frage:

Schüler weiß nicht, was  $a, b, x \in \mathbb{Q}$  bedeutet.

„Was bedeutet das Q in der Aufgabe oben, komme da nicht weiter...“

### Ablauf:

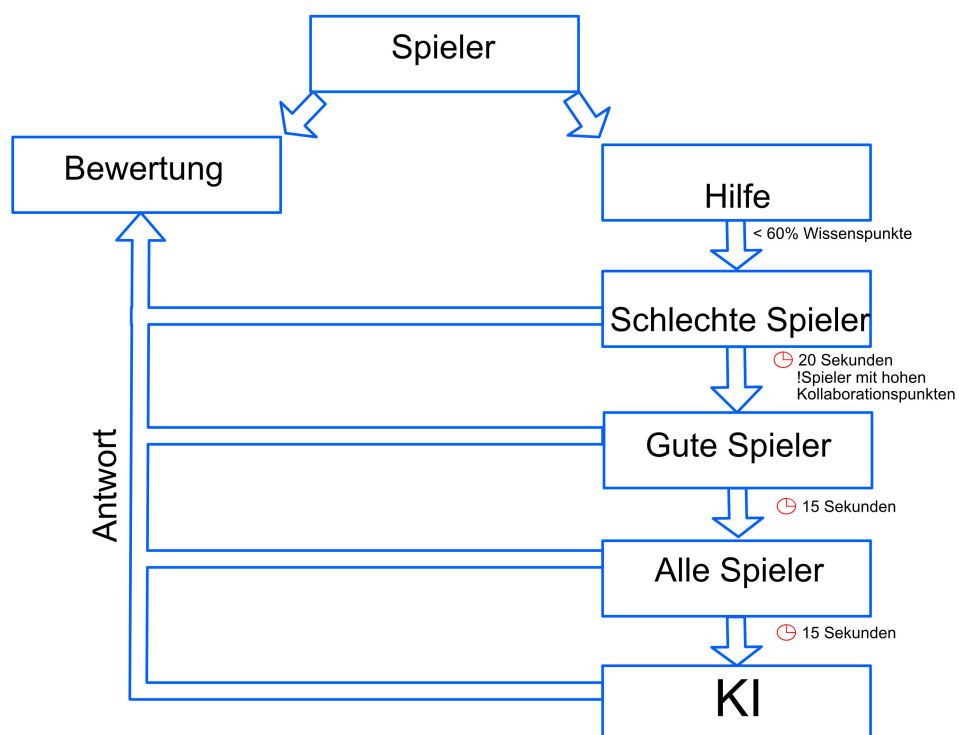


Abbildung 8: Ablauf der Fairness-Regelung

**Screendesign**

**User Stories**

## 6 Backup- / Restore-Vorgang

### Erstellen eines Backups

- 1) Einloggen in „vSphere Web Client (Flash)“
- 2) Auswahl der virtuellen Maschine im Navigator
- 3) Rechtsklick auf die virtuelle Maschine
- 4) "Snapshot" wählen
- 5) "Snapshot erstellen" wählen
- 6) Namen des Snapshots einstellen
- 7) Beschreibung des Snapshots einstellen
- 8) „Snapshot des Arbeitsspeichers der virtuellen Maschine erstellen“ auswählen
- 9) Eingabe bestätigen

Wird der Snapshot zum ersten Mal erstellt wird ein komplettes Abbild der virtuellen Maschine inklusive Arbeitsspeicher erstellt. Die Speicherung des Arbeitsspeicherinhaltes ist in sofern wichtig, dass es die Wahrscheinlichkeit eines fehlerfreien, konsistenten Wiederherstellens erhöht, falls Dateien bei der Erstellung des Snapshots geöffnet und nicht geschlossen sind.

Beim Erzeugen weiterer Snapshots wird nur das Delta zwischen den beiden Snapshots gespeichert, was zu reduzierter Speicherplatzverwendung führt.

Backups sollen entweder zu regelmäßigen Zeitpunkten (z.B. täglich oder an festgelegten Wochentagen) oder vor Einspielen eines Updates erstellt werden.

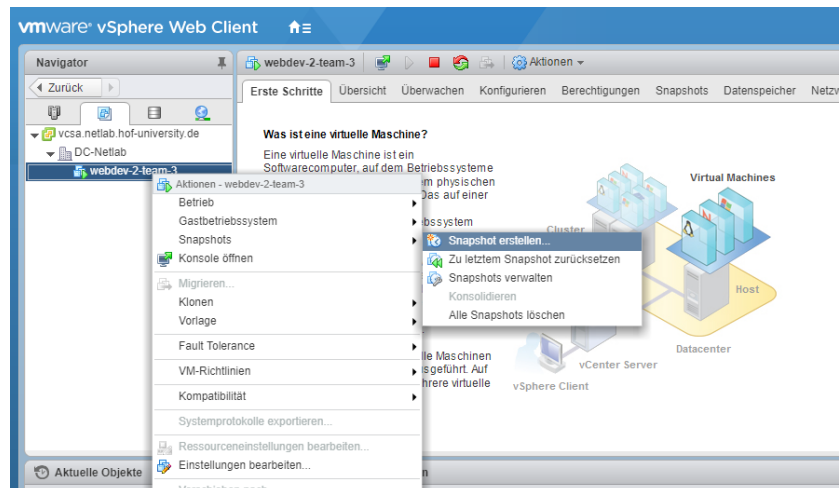


Abbildung 9: Erstellen eines Backups (Screenshot, Andreas Khouri)

## Wiederherstellen eines Backups

- 1) Einloggen in „vSphere Web Client (Flash)“
- 2) Auswahl der virtuellen Maschine im Navigator
- 3) Auswahl des Reiters "Snapshots"
- 4) Auswahl des wiederherzustellenden Snapshots
- 5) "Alle Aktionen" auswählen
- 6) "Wiederherstellen" auswählen
- 7) Sicherheitsfrage im folgenden Feld mit "Ja" beantworten

Falls bei der Wiederherstellung „Anhalten der virtuellen Maschine“ ausgewählt wurde, muss diese nach erfolgreichem Wiederherstellen wieder gestartet werden.

Ein Wiederherstellen eines Backups wird dann notwendig, wenn die virtuelle Maschine Inkonsistenzen aufweist oder ein eingespieltes Update rückgängig gemacht werden soll.

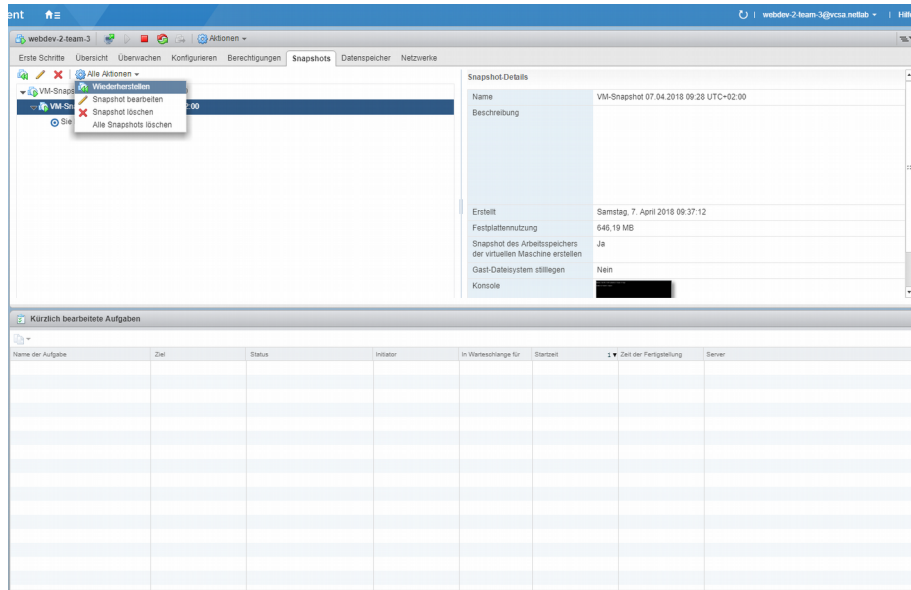


Abbildung 10: Wiederherstellung eines Backups (Screenshot, Andreas Khouri)

Dadurch, dass mehrere Wiederherstellungspunkte vorhanden sind, ist es möglich den Wiederherstellungszeitpunkt auch früher als das letzte Backup auszuwählen.

## Backup-Strategie

Wir haben uns dafür entschieden zu Beginn des Projekts und anschließend ab Beginn der Implementierungsphase täglich um 18.00 Uhr Backups zu erstellen, da Backup-Verwalter und Entwickler so ohne Leerlaufzeit unabhängig voneinander agieren können. Außerdem benötigt das Erstellen eines Backups keinen großen Zeitaufwand, da diese lokal auf dem Server der Hochschule Hof gespeichert werden. Zudem führt ein kurzes Backup-Intervall zu möglicher Arbeitersparnis nach einer Wiederherstellung.

Backup-Strategie	
Backup-Intervall:	Täglich

<b>Intervall-Beginn:</b>	<i>Beginn der Implementierungsphase</i>
<b>Intervall-Ende:</b>	<i>Fertigstellung des Projekts</i>
<b>Backup-Zeitpunkt:</b>	<i>18.00 Uhr</i>
<b>Speicherort:</b>	<i>Server der Hochschule Hof</i>
<b>Verantwortlicher:</b>	<i>Andreas Khouri</i>



## 7 Termine und zeitliche Abläufe

In diesem Kapitel wird kurz dargestellt, wann das Team sich für Meetings getroffen hat und was der thematische Inhalt dieser Meetings war.

### Meetings und Teamtreffen

Bezeichnung	Datum	Inhalt
Meeting 1	22.03.'18	In diesem Meeting wurden die Teams gebildet, es wurde ein allgemeiner Ausblick auf die Aufgaben gegeben und die ersten Vorbereitungen wurden getroffen.
Meeting 2	23.03.'18	Hier wurden den Teams im Allgemeinen das agile Vorgehen in Projekten vermittelt. Gezeigt wurden hier die nötigen Planungen und die Dokumente, die angefertigt werden könnten, um den Überblick über die Planung zu behalten.
Meeting 3	26.03.'18	Das erste Meeting für Team 3 in voller Besetzung. Es wurden Aufgabenbereiche und Zuständigkeiten geklärt. Diese werden in Kapitel 3 genauer beschrieben. Außerdem wurde eine Entscheidungsmatrix erstellt, die die Benutzung der Module in HumHub als Thema hatte.
Meeting 4	05.04.'18	Es wurde die allgemeine Intention des Projektes wiederholt, Hierfür hat Prof. Dr. Jürgen Heym für alle Teams die Regelsätze klar definiert, nach denen sich die Teams

		untereinander Ideen für das Spiel sammeln sollen.
Meeting 5	06.04.'18	Diesem Meeting wurde genauer auf das zu benutzende Framework Bootstrap eingegangen, eine Kurzeinführung wurde den Studenten nahegelegt.
Meeting 6	06.04.'18	Das Team hat die Regelsätze auf 2er Gruppen verteilt. Diese sollen sich Ideen und Maßnahmen für die beschriebenen Probleme überlegen.
Meeting 7	10.04.'18	Das Team hat sich getroffen, um seine Ergebnisse bei der Entwicklung der Regelsätze zu besprechen und gegebenenfalls abzuändern.
Meeting 8	12.04.'18	Es wurden die ersten Züge des Frameworks AngularJS vorgestellt
Meeting 9	13.04.'18	Der letzte Teil von AngularJS wurde vorgestellt, es wurde gezeigt, wie die Angular-Anwendungen in die HTML Seiten eingefügt werden sollen. Meetings für den Sprint Review mit Prof. Dr. Heym wurden festgelegt.
<b>Ende Sprint 1</b>		
Meeting 10	16.04.'18	Absprache und Abstimmung mit Auftraggeber über Dokumentation und Projektverlauf

## 8 Verweise und Quellen

### Abbildungsverzeichnis

Abbildung 1: Erstellen eines Backups (Screenshot, Andreas Khouri).....	12
Abbildung 2: Wiederherstellung eines Backups (Screenshot, Andreas Khouri).....	13
Abbildung 3: Markdown (Quelle fehlt).....	16
Abbildung 4: Markdown (Quelle fehlt).....	16
Abbildung 5: Bsp. halb richtige Antwort (Beispiel von Christa Strauß und Tina Amann).....	21
Abbildung 6: Bsp. leichte Aufgabe.....	22
Abbildung 7: Bsp. schwere Aufgabe.....	22
Abbildung 8: Bsp. Fragen-Antwort-Aufgabe mit Rechenweg.....	23
Abbildung 9: Smileys zur Bewertung der Hilfe.....	38
Abbildung 10: Ablauf der Fairness-Regelung.....	41

### Tabellenverzeichnis

Rollenverteilung.....	7
Backup-Strategie.....	13
Markdown-Syntax.....	15
Entscheidungsmatrix.....	18
Entscheidungsmatrix(Bewertungen).....	19
Meetings/Teamtreffen.....	42
Aufgabenverteilung(Sprint1).....	45
Unterschriftensammlung(Selbstarbeitserklärung).....	48

### Abbildungsverzeichnis:

Abb.1:	selbst ausgedachtes Beispiel
Abb.2:	Gerhard Steidle (1975), Einführung in die Algebra, Bayerischer Schulbuch-Verlag, ISBN 3-7627-3136-5, Seite 33. (Zahlen geändert)
Abb.3, 4:	Gerhard Steidle (1975), Einführung in die Algebra, Bayerischer Schulbuch-Verlag, ISBN 3-7627-3136-5, Seite 46. (Zahlen geändert)
Regelsatz 3	
Bild von Dzianis Brysiuk	(Affinity Designer)

## 9 Anhang

Im Anhang sind für den Prototypen des kollaborativen Spieles irrelevante Dokumente, die aber sehr wohl für den Ablauf des Projektes wichtig sind, aufgeführt. Zu diesen gehören im allgemeinen Projektmanagement-Dokumente, wie die Sprintplanungen, das Project-Backlog, eine Risikoanalyse, ein Backup-/ und Restoreplan und die Dokumentationen der für den Prototypen notwendigen Installationen auf dem Server.

### Sprintplanungen

Hier wird darauf eingegangen, wie die einzelnen Sprints geplant und durchgeführt wurden. Es werden die Aufgaben aufgelistet, den Teammitgliedern zugeordnet, zeitlich abgeschätzt und anschließend priorisiert. Abschließend wird jeder Sprint nochmals in der Retrospektive bewertet.

#### Erster Sprint

Dieser initiale Sprint ist für Grundlegende Aufgabenverteilung und Sammlung von Ideen bezüglich des kollaborativen Spieles zuständig. In ihm werden die Zuständigkeiten der Teammitglieder für elementare Teile des Projektes festgelegt.

#### Aufgaben

Nr	Bezeichnung	Bearbeiter	Fällig bis	Geschätzter Aufwand	Wirklicher Aufwand	Priorität (0<5)
1	Rollenverteilung	Team	26.03.'18	1h	1h	4
2	GitHub Repository erstellen, Einladungen versenden	Andreas G.	29.03.'18	1h	0,5	5

3	Trello einrichten, Einladungen versenden	Andreas G.	29.03.'18	1h	1h	5
4	Testen Zugriff auf Systeme	Dzianis, Noah	29.03.'18	0,5h	0,5h	5
5	Backup und Restore Plan erstellen	Andreas K.	02.04.'18	1,5h	1,5h	3
6	Installation von HumHub	Andreas K., Dzianis, Noah	05.04.'18	1h	1h	5
7	Def. Regelsatz 1	Christa, Tina	10.04.'18	2*2,5h	2*2,5h	5
8	Def. Regelsatz 2	Andreas G. Andreas K.	10.04.'18	2*2,5h	2*3h	5
9	Def Regelsatz 3	Dzianis, Noah	10.04.'18	2*2,5h	2*3h	5
10	Einarbeiten in Projektframeworks (Microsoft Project)	Dzianis, Noah	16.04.'18	3h	offen	
11	Risikoplan fertigstellen	Noah	19.04.'18	3h	offen	
12	Changelog fertigstellen	Noah	19.04.'18	2h	offen	

## Rückblick

In diesem Sprint wurden einige wichtige Aufgaben erledigt, das Team hat untereinander die Rollenverteilung ausgemacht und es wurden die ersten Dokumente zum festhalten des Projektverlaufes angefertigt. Rückblickend ist die strukturiert Planung des Verlaufes allerdings noch besser zu bewältigen.

Einige Kennzahlen sollten gesammelt werden, nach denen Übersichten erstellt werden können, um beim Abschluss eines Sprints mit Prof. Dr. Heym die Fortschritte besser evaluieren zu können. Positiv zu bewerten ist die Arbeitseinstellung der einzelnen Teammitglieder, die ihre Aufgaben bis jetzt ausnahmslos ohne Verzug erledigt haben. Demnach ist der Projektverlauf zu diesem Zeitpunkt absolut im Soll, der Abschluss der Projektes ist momentan nicht gefährdet.

## 10 Erklärung zur Selbstarbeit

### Erklärung

Ich erkläre hiermit, dass ich meinen Beitrag zur vorliegenden Gruppenarbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; das gleiche gilt für die von den auf dem Titelblatt der Arbeit genannten Autoren gemeinsam verfassten Teile. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde nach meiner besten Kenntnis bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Hof, den \_\_\_\_\_

Andreas Glaser,

\_\_\_\_\_

Andreas Khouri,

\_\_\_\_\_

Dzianis Brysiuk,

\_\_\_\_\_

Christa Strauß,

\_\_\_\_\_

Noah Lehmann,

\_\_\_\_\_

Tina Amann,

\_\_\_\_\_