



# **NoRPG: Entwicklung einer spielbasierten Lernspielplattform**

## **STUDIENARBEIT T2\_3201**

des Studiengangs Angewandte Informatik  
an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Mehmet Ali Incekara & Tom Wolske**

Abgabedatum 21. März 2017

<b>Bearbeitungszeitraum</b>	26 Wochen
<b>Matrikelnummer - Mehmet Ali Incekara</b>	5672336
<b>Matrikelnummer - Tom Wolske</b>	1156973
<b>Kurs</b>	TINF14B2
<b>Gutachter der Studienakademie</b>	Prof. Dr. Kay Berkling

# Erklärung

Gemäß §5 (2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 18. Mai 2009 erkläre ich hiermit,

1. dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.
2. dass die Übernahme von Zitaten und Gedankengut anderer Autoren gekennzeichnet wurde.
3. dass die eingereichte elektronische Fassung exakt mit der schriftlichen übereinstimmt.
4. dass ich die Projektarbeit keiner externen Prüfung vorgelegt habe.

Karlsruhe, den 21. März 2017

Ort, Datum

Tom Wolske

Karlsruhe, den 21. März 2017

Ort, Datum

Mehmet Ali Incekara

## **Abstract**

# **Zusammenfassung**

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>VI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Aufbau der Arbeit . . . . .	2
1.3 Ziel der Arbeit . . . . .	2
<b>2 NoRPG</b>	<b>3</b>
2.1 Gedanken . . . . .	4
2.2 Story . . . . .	5
2.3 Abgrenzung zu MOOC . . . . .	6
<b>3 Software Requirements Specification</b>	<b>8</b>
3.1 Einführung . . . . .	8
3.1.1 Zweck . . . . .	8
3.1.2 Umfang . . . . .	8
3.2 Allgemeine Beschreibung . . . . .	9
3.2.1 Produktperspektive . . . . .	9
3.2.2 Produktfunktionen . . . . .	10
3.2.3 Benutzermerkmale . . . . .	11
3.2.4 Einschränkungen . . . . .	11
3.2.5 Annahmen und Abhängigkeiten . . . . .	12
3.2.6 Aufteilung der Anforderungen . . . . .	12
3.3 Spezifische Anforderungen . . . . .	12
3.3.1 Externe Schnittstellen . . . . .	13
3.3.2 Funktionale Anforderungen . . . . .	18
3.3.3 Performanz Anforderungen . . . . .	27
3.3.4 Datenbank Anforderungen . . . . .	28
3.3.5 Entwurfsbeschränkungen . . . . .	29
3.3.6 Benutzerfreundlichkeit . . . . .	30
3.3.7 Zuverlässigkeit . . . . .	31
3.3.8 Verfügbarkeit . . . . .	31
3.3.9 Sicherheit . . . . .	32

3.3.10	Wartbarkeit . . . . .	32
3.3.11	Portabilität . . . . .	33
<b>4</b>	<b>Technische Grundlagen</b>	<b>34</b>
4.1	Unity . . . . .	34
4.2	Visual Studio . . . . .	36
4.3	C Sharp . . . . .	36
4.3.1	Allgemeiner aufbau C# . . . . .	37
4.3.2	Unity Skripte . . . . .	38
4.4	SQL . . . . .	38
<b>5</b>	<b>Umsetzung</b>	<b>39</b>
5.1	App . . . . .	39
5.2	Datenbank auf dem Handy . . . . .	39
5.3	Datenbank auf dem Server . . . . .	39
5.4	C# Skripte . . . . .	40
5.4.1	Kamera . . . . .	40
5.4.2	Player . . . . .	40
5.4.3	Portale . . . . .	40
5.4.4	Datenimport aus JSON . . . . .	40
5.4.5	Datenimport aus / in Datenbank . . . . .	40
5.4.6	Allgemein . . . . .	42
<b>6</b>	<b>Fazit und Ausblick</b>	<b>43</b>
6.1	Fazit . . . . .	43
6.2	Ausblick . . . . .	43
	<b>Anhang</b>	<b>IX</b>

# **Abkürzungsverzeichnis**

# Abbildungsverzeichnis

1	High-Level-View von NoRPG . . . . .	9
2	Mockup: Login-Screen . . . . .	14
3	Mockups: Registrierungsformular und Charaktererstellung . . . . .	15
4	Mockup: Head-Up Display . . . . .	15
5	Mockup: Menü . . . . .	16
6	Mockups: Spielliste, Fortschrittanzeige, Karte und Erfolgsübersicht . . .	16
7	Overall Use Case Diagramm . . . . .	18
8	Activity Diagramm: Create Character . . . . .	19
9	Activity Diagramm: Open Map . . . . .	20
10	Activity Diagramm: Show Games . . . . .	21
11	Activity Diagramm: View Progress . . . . .	22
12	Activity Diagramm: Achievements . . . . .	23
13	Activity Diagramm: Character Control . . . . .	25
14	Activity Diagramm: Game Interaction . . . . .	27
15	Darstellung von Tablemappings . . . . .	35



# Listings

1	Hello World in C# . . . . .	37
2	Aufbau eines Unity Skriptes . . . . .	38
3	Skript SendDataToServer.cs . . . . .	41

# 1 Einleitung

Spiele, jeder kennt sie und spielt regelmäßig welche. Sie sind ein fester Bestandteil unserer Kultur und das schon seit tausenden Jahren. Die ersten Gesellschaftsspiele wurden noch im Sand mit Stöcken oder Steinen gespielt. Eines der frühesten Spiele wird auch heutzutage noch gespielt, dabei handelt es sich um Mühle, ein Spiel welches die Ägypter vor bereits 3000 Jahren gespielt haben.<sup>1</sup> Spiele haben sich seit dem jedoch weiterentwickelt und dienen heutzutage nicht nur zum munteren Zeitvertreib.

Ob als Brett, Karten oder Glücksspiel, Spiele sind überall zu finden und jeder kann sie spielen. Seit 1972 entwickeln sich darüber hinaus weitere Spiele, Videospiele. Sie nutzen die immer größer werdende Rechenleistung von Computern aus, um uns immer realistischer aussehender Spiele zu liefern. Um den Überblick über die Vielzahl an Videospiele zu behalten, haben sich in den letzten Jahren verschiedene Plattformen etabliert, die versuchen dem Nutzer das zu bieten, was sie suchen. Dabei bieten diese viele verschiedene Arten von Spielen an, die einen beim Spielen die Zeit vergessen lassen. Beispiele für solche Plattformen sind unter anderem Steam, Uplay oder Origin.

Allerdings können Spiele uns nicht nur die Zeit vergessen lassen und für heitere Stunden sorgen, sie können uns auch Wissen vermitteln. Sei es beispielsweise durch eine Geschichte die sich real abgespielt hat, wie der erste Weltkrieg. Dieses Wissen wird unterbewusst an den Nutzer vermittelt, ohne das er aktiv versucht dieses zu lernen.

Für diesen Zweig hat sich eine eigene Branche entwickelt, welche sich mit Lernspielen befasst und versucht uns, über Videospiele, dieses Wissen zu vermitteln. Viele dieser Spiele nutzen bekannte Figuren, welche die Kinder aus dem Fernsehen kennen, um dieses Wissen zu vermitteln.

Diese Spiele werden hauptsächlich in den Schulen eingesetzt, um den Kindern Wissen spielerisch zu vermitteln. Jedoch profitiert nicht jedes Kind von diesem Vorteil. Sei es, weil die Schule keine Computer hat, oder weil das Kind nicht zur Schule gehen kann. Für diesen Zweck wurde die Plattform Hone<sup>2</sup> entwickelt, mit der Kinder, die nicht zur Schule gehen können, die Möglichkeit haben, Wissen zu erlangen.

---

<sup>1</sup> vgl. gesellschaftsspiele.de [spiele] (2015)

<sup>2</sup> siehe <http://hone-kids.herokuapp.com/>

## **1.1 Motivation**

Bei Hone handelt es sich um eine Spielplattform auf der sich Kinder, bevorzugt aus Regionen in denen Bildung mangelhaft ist, anmelden können. Auf dieser Plattform haben Sie dann eine Ansicht Ihrer, durch Spiele gelernte Kompetenzen, bzw. können sich die Kinder dort neue Spiele herunterladen, um weitere Kompetenzen zu erwerben. Das Aussehen und die Bedienung dieser Plattform ist allerdings nicht reizvoll für Kinder gestaltet.

Deshalb soll eine mobile App entwickelt werden, in der die Kinder auf spielerischer Weise Fortschritte machen. Dabei werden die Inhalte von Hone nicht übernommen und die App wird unabhängig von Hone funktionieren. Durch die Umsetzung als App gelingt es darüber hinaus den Kindern eine Plattform anzubieten, welche sie unabhängig von der Internetverbindung nutzen können. Daruch bauen die Kinder darüber hinaus ihre technischen Fähigkeiten im Umgang mit dem Smartphone aus.

## **1.2 Aufbau der Arbeit**

Die Arbeit beginnt mit der grundlegenden Idee Hinter der App. Darauf folgend wird die Spezifikation durchgeführt und dargestellt. Anschließend wird auf die technischen Grundlagen eingegangen, in denen die verwendeten Technologien und auf die umzusetzenden Ziele vorgestellt werden, damit verstanden werden kann was getan wurde. Darauf aufbauend wird die Implementierung der zuvor erklärten Ziele erläutert und es wird auf schwierige Stellen in der Umsetzung eingegangen. Abgeschlossen wird diese Arbeit mit einem Fazit und einem kurzen Ausblick, in dem der weitere Werdegang des Projektes geschildert wird.

## **1.3 Ziel der Arbeit**

Die Arbeit hat als Ziel eine App zu gestalten, welche es Kindern spielerisch möglich macht, Lernfortschritte zu erzielen. Die App wird dabei für Smartphones mit dem Betriebssystem Android optimiert.

Das Ziel dieser Arbeit ist es dabei eine Dokumentation über die Vorgehensweise zu liefern, sowie eine Dokumentation, mit deren Hilfe andere arbeiten können.

## 2 NoRPG

Bei NoRPG handelt es sich um ein Spiel, welches an ein Role Player Game (RPG) erinnern soll. Bei RPGs spielt der Spieler einen Charakter in einer fiktiven Welt. Dabei gibt es eine Story die während des Spielens erlebt wird. Damit diese Geschichte voran geht, muss der Spieler verschiedene Missionen bzw. Quests erledigen, um einen Fortschritt im Spiel zu erlangen. Dabei handelt es sich um die verschiedensten Aufgaben. Darüber hinaus sammelt der Spieler Objekte in der Welt, welche er anschließend im Spiel nutzen kann. Ein Beispiel für solche Spiele ist das Spiel The Witcher 3, welches auf diesen Prinzipien aufbaut.

Darüber hinaus gibt es noch sogenannte MMORPGs, Massive Multiplayer Online Role Player Game. Dabei gibt es wie bei RPGs eine Story und Quests, allerdings kann man auch auf andere Spieler treffen und mit diesen gemeinsam spielen. Das wohl berühmteste MMORPG ist dabei World of Warcraft vom Entwickler Blizzard. Hier kann der Spieler zwischen verschiedenen Klassen einen Charakter auswählen, von menschenähnlichen bis zu skurrilen Gestalten. Jede dieser Klassen hat verschiedene Fähigkeiten, welche sich auf den Spielverlauf auswirken. Darüber hinaus gibt es in MMORPGs verschiedene Events, an denen die Spieler gemeinsam versuchen eine Quest zu erfüllen. Dieser Multiplayermodus grenzt die MMORPGs von den RPG ab.

NoRPG hat keine Multiplayer Möglichkeiten und wird deswegen nur als ein RPG klassifiziert. Allerdings muss der Spieler keine Quests erfüllen, wie sie aus anderen Spielen bekannt sind. Der Nutzer muss andere Spiele spielen, damit der Spieler voran schreiten kann. Deshalb wurde sich für den Namen NoRPG entschieden, da es sich bei dem Spiel um ein RPG handelt, aber nicht alle klassischen Eigenschaften eines RPGs besitzt.

NoRPG ist ein Spiel, welches versucht Bildung für jeden erreichbar zu machen. Dieses Ziel ist dabei in den Global Goals definiert. Dabei handelt es sich um 17 Ziele welche bis 2030 Umgesetzt werden sollen, um das Leben für alle Menschen auf der Welt zu verbessern<sup>3</sup>. 2015 haben 193 Weltführer diese Unterzeichnet und begonnen dieses umzusetzen. Dabei sind diese Ziele umfangreich und reichen von einem besseren Umgang mit den uns zur Verfügung stehenden Ressourcen bis hin zu qualitativ hochwertiger Bildung für jeden und kostenlos.

---

<sup>3</sup> vgl. Global Goals [global] (2017)

NoRPG unterstützt dabei das Ziel hochwertige Bildung für jeden zugänglich zu machen, da die App für jeden frei, kostenlos und überall angeboten wird. Dieses Ziel hat weitere Unterziele, wobei nun kurz auf die für NoRPG relevanten Unterziele eingegangen wird.

- Bis 2030 sicherstellen, dass alle Mädchen und Jungen gleichberechtigt eine kostenlose und hochwertige Grund- und Sekundarschulbildung abschließen, die zu brauchbaren und effektiven Lernergebnissen führt.

Dieses Unterziel wird in NoRPG dahingegen unterstützt, dass die Common Core State Standards implementiert werden. Dies sind Standards für Unterrichtsfächer und beschreiben den zu lernenden Inhalt für Kinder in den verschiedenen Klassen. Da NoRPG für alle kostenfrei zugänglich ist und Mädchen und Jungen gleichberechtigt sind, werden auch diese zwei Aspekte des Unterziels unterstützt. Die Standards werden dabei von den zu Verfügung gestellten Spielen erfüllt.

- Aufbau und Weiterentwicklung von Bildungseinrichtungen, die kinder- und behindertengerecht und geschlechtsspezifisch sind und für alle eine sichere, gewaltfreie, integrative und effektive Lernumgebung bieten.

Da NoRPG keine Bildungseinrichtung ist wird dieses Unterziel nur bedingt erfüllt. NoRPG bietet Kindern jedoch eine sichere, gewaltfreie, integrative und effektive Lernumgebung, wodurch dieses Ziel allerdings zum Teil erfüllt wird. Darüber hinaus ist diese Umgebung für Kinder ausgelegt. Geschlechtsspezifisch ist das Spiel nur dahingehend, dass die Kinder zu Beginn das Geschlecht ihres Charakters auswählen.

## 2.1 Gedanken

Dieses Ziel der Global Goals wird dabei durch Gamification umgesetzt. Bei Gamification oder Gamifizierung handelt es sich um das Nutzen von spielespezifischen Eigenschaften, welche außerhalb von Spielen angewendet werden. Ein Beispiel dafür ist PayBack. Dabei sammeln die Nutzer bei jedem Einkauf Punkte. Diese Punkte können die Kunden dann gegen Prämien eintauschen. In Videospielen sammeln die Spieler zum Beispiel Münzen um diese anschließend einzutauschen.

Gamification setzt darüber hinaus auch weitere erfolgreiche Prinzipien aus Videospielen um, damit die Nutzer Spaß daran haben, das Produkt zu nutzen. Dadurch soll die Motivation gesteigert werden, NoRPG zu verwenden. Beispiele für weitere Praktiken die Gamification einsetzen:

- Einbettung in eine Geschichte
- direktes Feedback
- Belohnungen
- Status durch Level und Auszeichnungen
- Wettbewerb
- Teamaktivität

Darüber hinaus sind die Faktoren wie Erfolgserlebnisse, Gruppenzugehörigkeit und soziale Akzeptanz wichtig.

## 2.2 Story

In NoRPG spielt der Spieler einen Charakter der zu Beginn der Geschichte in einem Dorf wohnt. Von diesem können der Spieler durch Portale in zunächst fünf andere Welten gehen, in denen andere Lebensbedingungen herrschen. Eines Tages wurde dieses Dorf von einem Bösewicht heimgesucht und dieser hat die komplette Welt, in der der Spieler lebt, farblos gemacht. Zusammen mit den Farben wurden die Emotionen aus dem Dorf genommen. Nun hat sich der Hauptcharakter das Ziel gesetzt, die Farben und die Emotionen der Einwohner zurück zu bringen, damit wieder alles wie vor dem Angriff ist. Dazu muss der Spieler in die verschiedenen Welten gehen und verschiedene Aufgaben erfüllen.

Die Welten sind dabei in fünf verschiedene Themengebiete aufgeteilt, Wald, Tropen, Wüste, Eis und Feuer. Die Welten sind jeweils wieder in fünf Bereiche gegliedert, die der Spieler mit der Zeit erreichen kann. Dabei wird der zu erkundende Bereich größer, je mehr Standards der Spieler erfüllt hat.

### **Startwelt: Dorf**

In die verschiedenen Welten gelangen kann der Spieler über Portale, welche in der Startwelt vorhanden sind. Jede der fünf Welten spiegelt eine Klasse wieder, so sind Standards der ersten Klasse eins in der ersten Welt. Äquivalent verhält es sich mit den anderen Welten.

### **Welt 1: Wald**

In dieser Welt ist Wald das primäre Element. Dieser ist in verschiedenen Ausprägungen vorhanden, von sehr Dicht bis hin zu Lichtungen.

### **Welt 2: Tropen**

Diese Welt erinnert an die Tropen. Es wird sehr bunt und es wird viel Tropenwald, jedoch auch viel Wasser geben. Diese Welt besteht dabei aus mehreren Inseln, welche über Brücken erreicht werden können.

### **Welt 3: Wüste**

In Welt 3 dreht sich alles um die Wüste. Diese besteht aus Dünen und ist trostloser wie die vorherigen Welten. Innerhalb dieser Wüste gibt es Oasen, verschiedene Ruinen und Pyramiden zu entdecken.

### **Welt 4: Eis**

In der Eisswelt dreht sich alles um Eis und Schnee. Es wird viele hohe Berge mit großen Höhlen geben.

### **Welt 5: Lava**

In Welt 5 dreht sich alles um Feuer. In dieser Welt gibt es verschiedene Inselplattformen, welche durch Lava getrennt sind. Diese sind untereinander mit Brücken verbunden. Auf den verschiedenen Inseln kann der Spieler verschiedene Dinge erkunden, darunter Drachen, Vulkane oder Ruinen.

## **2.3 Abgrenzung zu MOOC**

NoRPG ist allerdings kein Massive Open Online Course (MOOC) sondern baut nur auf einem auf. Die Daten dieses MOOCs werden in NoRPG zur Verfügung gestellt, damit die Kinder einen Anreiz haben, diesen zu nutzen. Der eigentliche MOOC ist dabei Hone. Dabei handelt es sich um eine Lernplattform, welche es für Kinder ermöglicht, grundlegendes Verständniss, welches in der Grundschule vermittelt wird, zu erlangen. Dabei stellt Hone den Kindern eine Liste von Spielen bereit. Durch spielen dieser Spiele

lernen die Kinder die Dinge die sie auch in der Grundschule lernen würden. Dabei sind diese Lerninhalte in den CCSS definiert und aufgegliedert.



## 3 Software Requirements Specification

Das Software Requirements Specification, kurz SRS, ist ein veröffentlichter Standard zur Spezifikation einer Software. Der Inhalt eines SRS ist vom Institute of Electrical and Electronics Engineers im Standard IEEE 830-1998 festgehalten.

Der Aufbau dieses Kapitels entspricht der Struktur, die in dem Standard beschrieben wird. Einige Kapitel des SRS werden allerdings nicht behandelt, da sie keine Relevanz für NoRPG haben oder an einer anderen Stelle in diesem Dokument erwähnt werden.

### 3.1 Einführung

Das erste Kapitel des SRS enthält eine Beschreibung und eine Übersicht über alles, was im SRS enthalten ist.

#### 3.1.1 Zweck

Das SRS beschreibt den kompletten Projektumfang und die Anforderungen an die Software NoRPG. Es illustriert den Zweck und die vollständige Erklärung für die Entwicklung der Software. Dabei werden unter anderem Systemeinschränkungen, Schnittstellen und Interaktionen mit externen Schnittstellen thematisiert<sup>4</sup>.

Die Zielgruppe des SRS bzw. die Stakeholder des Projektes sind alle Personen und Personengruppen, die in irgendeiner Verbindung mit NoRPG stehen oder jene, die Interesse an der Umsetzung haben<sup>5</sup>. Zudem dient die Spezifikation zur Kommunikation zwischen den Stakeholdern und den Entwicklern.

#### 3.1.2 Umfang

Dieses SRS handelt von der in Kapitel 2 beschriebenen Software NoRPG.

---

<sup>4</sup> vgl. Tripp [srsIEEE](1998) Seite 3

<sup>5</sup> vgl. Rozanski [rozanski2011](2011) Seite 6

## 3.2 Allgemeine Beschreibung

Im zweiten Kapitel des SRS werden allgemeinen Faktoren, die das Produkt und seine Anforderungen betreffen, beschrieben. Dieses Kapitel bietet einen Überblick über die Systemfunktionalitäten und stellt verschiedene Arten von Stakeholdern und deren Interaktionen mit dem System vor. Dieses Kapitel behandelt jedoch nicht die spezifischen Anforderungen, sondern stellt den Hintergrund für diese dar.

### 3.2.1 Produktperspektive

Das zu beschreibende vollständige System NoRPG besteht aus mehreren Komponenten, die auf unterschiedlichste Weise mit den Stakeholdern kommunizieren. Daher ist es besonders wichtig, das Produkt in unterschiedlichen Perspektiven mit verwandten und geplanten Produkten zu betrachten. Aus diesem Grund werden alle System-, Benutzer-, Hardware- und Softwareschnittstellen von NoRPG definiert.

Folgende Grafik 1 stellt dabei die High-Level-View von NoRPG und seinen Komponenten dar.

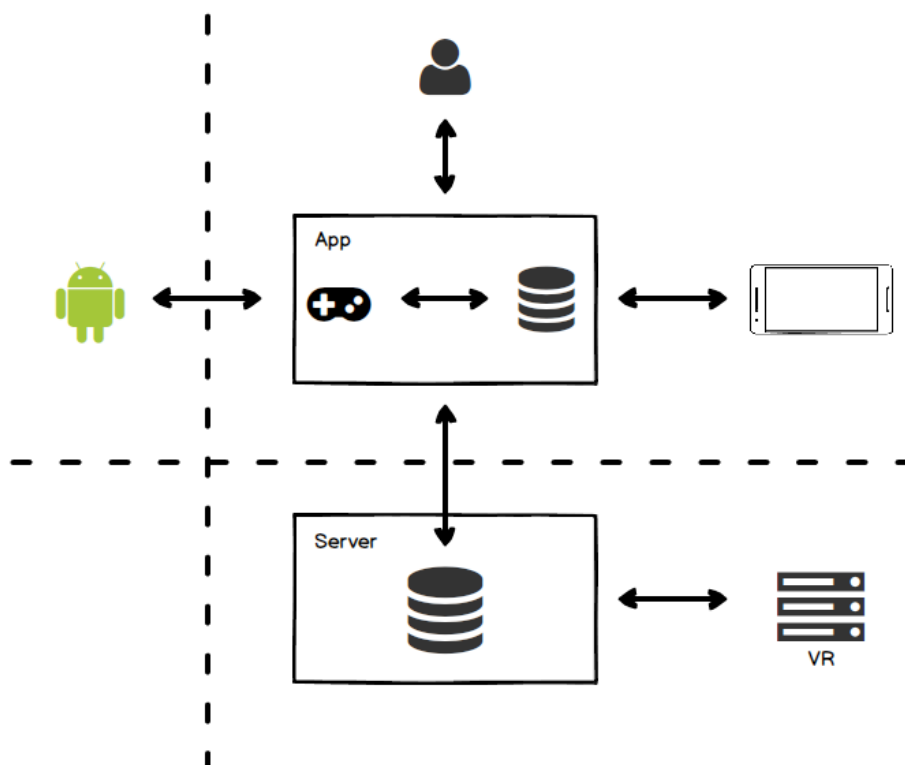


Abbildung 1: High-Level-View von NoRPG

Das vollständige System von NoRPG besteht aus zwei Kernkomponenten. Die erste Kernkomponente ist die Android App, welche sich aus dem Quellcode des Spieles und

einer eingebetteten lokalen Datenbank zusammensetzt. Die zweite Kernkomponente ist eine Datenbank, die sich auf einem Server befindet.

Die Schnittstelle zwischen der App und dem Betriebssystem des Smartphones ist eine Systemschnittstelle. Systemschnittstellen identifizieren die Funktionalität der Software, um die Systemanforderung und Schnittstellenbeschreibung zu erfüllen, damit die Software mit dem System übereinstimmt<sup>6</sup>.

Bei der App NoRPG handelt es sich um die Benutzerschnittstelle des Systems. Benutzerschnittstellen beschreiben die Kommunikation zwischen dem System und dem User. Die Benutzeroberfläche der App ist die einzige Möglichkeit für den Anwender mit dem System zu interagieren.

Jede Schnittstelle zwischen NoRPG und Hardwarekomponenten des Systems werden als Hardwareschnittstellen bezeichnet. Das Smartphone mit all seinen Komponenten sind Hardwarekomponente, zu der eine direkte Schnittstelle existiert. Die Hardwarekomponenten eines Smartphones sind der Touchscreen, die Lautsprecher oder der WLAN-Adapter. Diese Komponenten werden in der Abbildung durch das Smartphone zusammengefasst. Eine weitere Hardwareschnittstelle gibt es zwischen dem Server und der Hardware, auf dem dieser installiert ist.

Die App kommuniziert mit der lokalen sowie mit der serverseitigen Datenbank und verwendet dabei die Funktionen von anderen Softwareprodukten. Dabei handelt es sich um Softwareschnittstellen. Sie bilden den Übergang zwischen unterschiedlichen Programmen und ermöglichen dadurch das Nutzen derer Funktionalitäten.

### **3.2.2 Produktfunktionen**

In diesem Unterkapitel werden die wichtigsten Funktionen von NoRPG zusammengefasst. Wie in Kapitel 2 beschrieben ist das Hauptziel von NoRPG Lernspiele in einer standardisierten Reihenfolge zum Herunterladen anzubieten. Dieses Ziel macht das Downloaden von Spielen zu der Hauptfunktionalität von NoRPG.

Neben dieser Funktion gibt es allerdings weitere Produktfunktionen, um NoRPG attraktiver zu gestalten und zu personalisieren. Der Spieler wird in der Lage sein mit Elementen im Spiel zu interagieren und dabei Spielgegenstände zu sammeln. Um dem Anwender eindeutig identifizieren zu können, wird NoRPG über eine eigene Anmelde- und Registrierungsfunktion verfügen. In diesem Registrierungsprozess ist es dem Spieler möglich, seinen eigenen persönlichen Charakter zu erstellen.

---

<sup>6</sup> vgl. Tripp [srsIEEE](1998) Seite 13

Im Spiel selbst wird es neben Spieloptionen wie Qualitäts- und Audioeinstellungen noch Features geben, die das Spielerlebnis verbessern sollen. Dazu zählen Funktionen wie das Öffnen einer Karte der aktuellen Spielwelt oder das Betrachten des Fortschritts im jeweiligen Standard.

Die App speichert den Fortschritt und die Daten in der lokalen eingebetteten Datenbank und synchronisiert diese Informationen mit dem Server.

### **3.2.3 Benutzermerkmale**

Im Rahmen dieser Studienarbeit wird zunächst nur eine Benutzergruppe vollständig implementiert und daher nur diese hier beschrieben.

Die zu implementierende Benutzergruppe sind die User, viel mehr die Spieler. Grundsätzlich richtet sich NoRPG an Kinder, die keine Möglichkeit haben eine Schule zu besuchen. Jedoch werden keine Benutzergruppen für diese App ausgeschlossen. Egal ob jung oder alt, männlich oder weiblich, der Spieler sollte nur eine Neugier zum Lernen mitbringen.

Der Anwender benötigt Erfahrung mit der Verwendung eines Smartphones, insbesondere mit einem Android-System. Dazu zählt die Bedienung der Android-Oberfläche und die des Google Play Stores. Zudem sollten die User englische Texte verstehen können, da NoRPG zunächst nur in der englischen Sprache erscheinen wird.

### **3.2.4 Einschränkungen**

Da das SRS für die Kommunikation zwischen Entwickler und Stakeholder dient, wird zwischen Einschränkungen für Entwickler und für Spieler unterschieden.

Grundsätzlich müssen sich Entwickler an die vorgegebenen regulatorischen Richtlinien, wie beispielsweise an die Datenschutzerklärung von Google oder an das IT-Sicherheitsgesetz halten.

Da NoRPG sich an Kinder in bildungsfernen Ländern richtet ist es besonders wichtig, dass die Texte in NoRPG einfach zu verstehen sind. Da das Spiel zunächst nur in Englisch erscheinen wird, dürfen die englischen Texte kein Fachjargon oder ähnliches beinhalten. Die App darf keine hohen Mindestanforderungen an Hardwareressourcen haben, da der technische Standard in bildungsfernen Ländern geringer ist. Das bedeutet für die Entwickler das Spiel so gut wie möglich ressourcenschonend umzusetzen. Des Weiteren gilt es bei der Implementierung zu beachten, dass NoRPG soweit wie möglich ohne eine aktive Internetverbindung spielbar bleiben muss.

Allerdings gibt es auch Einschränkungen, welche für die Spieler gelten oder zumindest temporär. Wie schon öfter erwähnt wurde, wird das Spiel zunächst nur in Englisch erscheinen. Dementsprechend benötigt der Spieler Englischkenntnisse um die Texte im Spiel lesen und verstehen zu können. Für die Anmeldung, die Registrierung, das Herunterladen von Spielen, das Synchronisieren und installieren von Updates wird eine aktive Internetverbindung vorausgesetzt. Des Weiteren benötigt der Spieler ein Android Smartphone, welches die Mindestanforderungen von NoRPG erfüllt.

### **3.2.5 Annahmen und Abhängigkeiten**

Eine Annahme von NoRPG ist, dass es immer auf Smartphones, die genügend Leistung haben, verwendet wird. Wenn das Telefon nicht über genügend Hardwareressourcen für die Anwendung verfügt, kann es Szenarien geben, in denen die Anwendung nicht wie beabsichtigt oder überhaupt nicht funktioniert.

Eine weitere Annahme ist, dass das Smartphone und dessen Hardware sowie Software funktionieren. Das Smartphone muss sich mit dem Internet verbinden können, wenn der Benutzer sich anmelden möchte oder Lernspiele herunterladen will. Neben einer funktionierenden Internetverbindung sollten andere Hardwareelemente wie die Lautsprecher oder der Touchscreen funktionieren. Das Smartphone muss eine gültige Android Version mit einem Google Konto besitzen.

### **3.2.6 Aufteilung der Anforderungen**

In dem Fall, dass das Projekt verzögert wird, gibt es einige Anforderungen, die auf die nächste Version der Anwendung übertragen werden könnten.

## **3.3 Spezifische Anforderungen**

Das letzte Kapitel des SRS dient dazu alle Anforderungen an die Software detailliert zu beschreiben. Dies ermöglicht es Entwicklern ein System zu entwickeln, welches allen Anforderungen entspricht, und Testern, NoRPG ausreichend zu testen.

### 3.3.1 Externe Schnittstellen

Dieser Abschnitt ist die detaillierte Beschreibung aller Ein- und Ausgänge von NoRPG. Diese Beschreibung ergänzt und vervollständigt die Schnittstellenbeschreibung von Kapitel 2.2.1.

#### Systemschnittstellen

NoRPG hat genau eine Schnittstelle mit einem anderen System und zwar mit Android. Android ist das Betriebssystem von Google für mobile Geräte, welches aktuell in der Version 7.0 Nougat zu erhalten ist. Viele Smartphone-Hersteller nutzen Android als Basis für ihr eigenes auf Android aufbauendes Betriebssystem.

Der Gültigkeitsbereich der Systemschnittstelle ist auf die App begrenzt und hat keinerlei direkte Auswirkungen auf den Server.

Das Datenformat von Android ist das Android Application Package (APK) und wird für die Distribution und Installation von mobilen Apps auf Android Smartphones verwendet. Eine APK-Datei enthält den gesamten Programmcode, Ressourcen, Assets, Zertifikate und Metadaten. Verglichen kann das Datenformat von Google mit einem ZIP-Archiv<sup>7</sup>. Dieses Format muss NoRPG erfüllen, um unabhängig von den zu implementierenden Produktfunktionen auf einem Android Smartphone laufen zu können.

#### Benutzerschnittstellen

Die Benutzerschnittstellen bzw. User Interfaces (UI) sind der Punkt, an dem die Benutzer mit der Software interagieren. Zur Beschreibung der Benutzerschnittstellen werden logische Eigenschaften und Aspekte zur Optimierung formuliert. Für die Veranschaulichung werden Mockups verwendet. Diese stellen dar, wie die Oberfläche aussehen kann. Die am Projektende implementierte Benutzeroberfläche kann sich von den Mockups unterscheiden.

Wenn der Benutzer NoRPG das erste Mal startet oder nicht angemeldet ist, wird ihm die Login-Oberfläche (siehe Abbildung 2) präsentiert. Auf dieser Oberfläche hat der Benutzer die Möglichkeit sich mit seinem Benutzernamen und Passwort anzumelden oder sich, falls noch nicht geschehen, bei NoRPG zu registrieren. Das Smartphone muss Quer gehalten werden, da alle Elemente des Bildschirms für diese Ausrichtung

---

<sup>7</sup> vgl. Dan Morrill (Google) [google1]

angeordnet sind. Diese Eigenschaft betrifft auch alle anderen Benutzerschnittstellen und wird nicht extra erwähnt.

Die Hauptelemente des Login-Screens sind die Eingabefelder für Benutzername und Passwort, der Login- und Registrierungsbutton sowie ein Ladebalken, der den aktuellen Status von NoRPG zeigt. Wenn das Spiel aktualisiert wird kann hier der Status abgelesen werden. Zur Optimierung der Nutzung ist das Layout der Login-Oberfläche ein Border-Pane, in dem die Bestandteile in einer einzigen Spalte angeordnet sind. Fehler werden in einem kleinen Fenster dargestellt, wenn beispielsweise der Benutzer falsche Login-Daten eingibt oder die Internetverbindung während des Aktualisierungsprozesses abbricht.

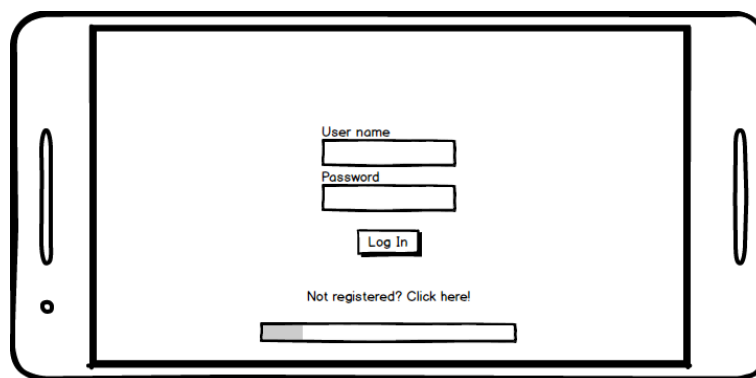


Abbildung 2: Mockup: Login-Screen

Falls sich der Benutzer bei NoRPG registrieren möchte, hat er die Möglichkeit dies direkt in der App zu machen. Dazu klickt der Benutzer den Register-Button auf der Login-Oberfläche. Anschließend öffnet sich die Register-Oberfläche. Der vollständige Registrierungsprozess (siehe Abbildung 3) setzt sich aus zwei Schritten zusammen. Im ersten Schritt muss der Spieler das Registrierungsformular ausfüllen und bestätigen. Die Elemente des Formulars sind als Tabellen-Layout angeordnet, welches die Lesbarkeit verbessert. Felder, die nicht der erwarteten Eingabe entsprechen, werden als Falsch markiert und visuell hervorgehoben. Wenn das Anlegen des Accounts erfolgreich war, hat der Anwender die Möglichkeit im zweiten Schritt seinen persönlichen Charakter zu erstellen. Dafür bestimmt der Benutzer den Namen, das Geschlecht und das Aussehen.

NoRPG startet erst, nachdem alles geladen wurde und der Benutzer angemeldet ist. Der Bildschirm von NoRPG besteht aus der Spielwelt (Grafik) und dem Head-Up Display (HUD). Das HUD ist eine Methode, mit der Informationen visuell als Teil der Benutzeroberfläche eines Spiels vermittelt werden. Während die Informationen, die auf dem HUD angezeigt werden, stark vom Spiel abhängen, gibt es viele Eigenschaften, die Spieler über viele Spiele erkennen. Die meisten von ihnen sind statisch auf dem Bildschirm, so dass sie während des Spiels sichtbar bleiben.

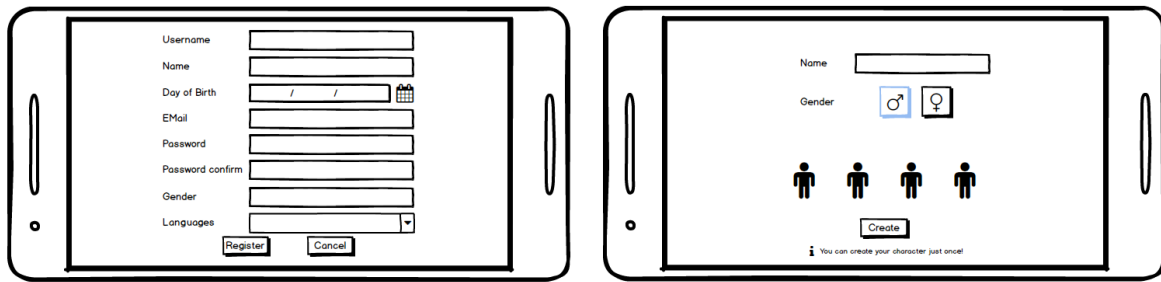


Abbildung 3: Mockups: Registrierungsformular und Charaktererstellung



Abbildung 4: Mockup: Head-Up Display

Das Mockup in Abbildung 4 enthält alle direkt sichtbaren HUD Elemente, die während des Spieles aktiv sind. Diese Elemente sind an die Ecken des Bildschirms gebunden, so befinden sich beispielsweise die Pfeiltasten zur Bewegung des Charakters in der linken unteren Ecke des Bildschirms und die Buttons zur Interaktion mit dem Spiel in der unteren rechten Ecke.

Das Menü (siehe Abbildung 5), welches sich in der oberen linken Ecke befindet, kann geöffnet werden. Dadurch verändert sich das HUD von NoRPG und es erscheinen neue Elemente, die der Spieler sehen und benutzen kann. NoRPG ist derweil pausiert. Die anderen Elemente hingegen werden überdeckt oder reagieren nicht solange das Menü offen ist. Deshalb ergeben sich neue Optionen bzw. Möglichkeiten für den Spieler um mit NoRPG zu interagieren.

Es wird zwischen zwei Typen von Menü-Funktionen unterschieden. Eine Funktionen können direkt im Menü durchgeführt werden, wie beispielsweise die Qualitätseinstellung. Einige der Funktionen wiederum öffnen ein Fenster bzw. eine neue Ansicht, in dem die neuen Funktionalitäten zur Verfügung stehen. Die Fenster müssen erst geschlossen werden um das Spiel fortsetzen zu können. Beispiele sind in der folgenden Abbildung 6 als Mockups zu betrachten.



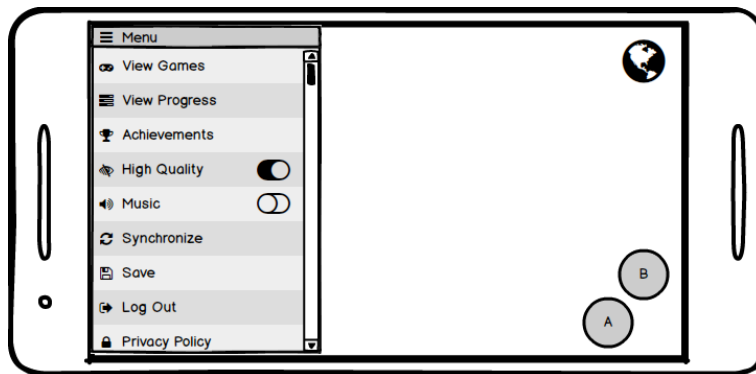


Abbildung 5: Mockup: Menü

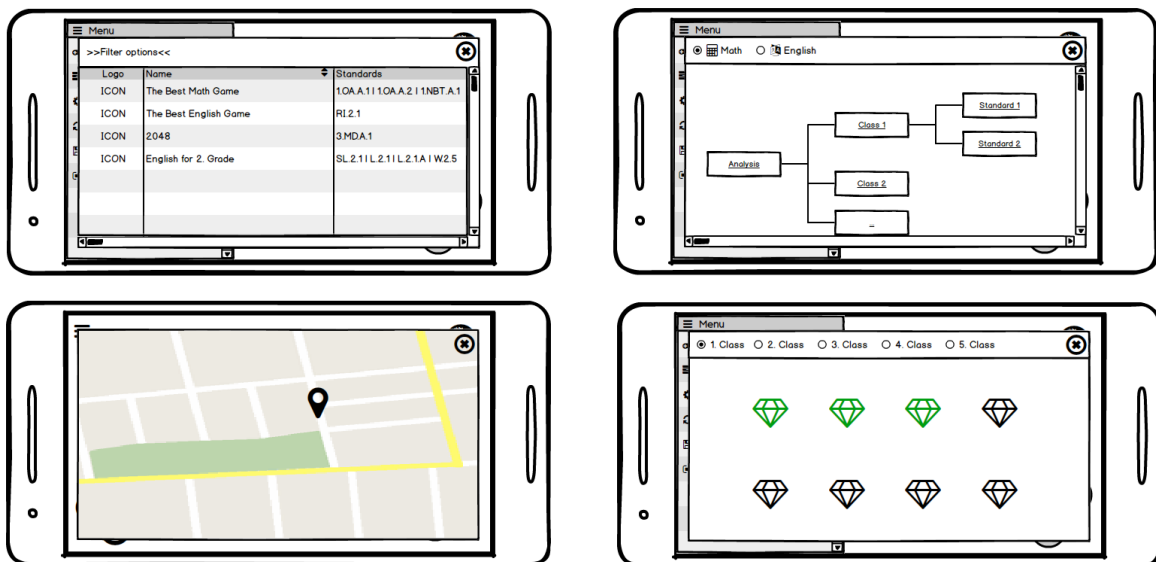


Abbildung 6: Mockups: Spielliste, Fortschrittanzeige, Karte und Erfolgsübersicht

## **Hardwareschnittstellen**

Schnittstellen zwischen NoRPG und Hardwarekomponenten werden als Hardware-schnittstellen des Systems bezeichnet. Dieses Kapitel dient zur Spezifikation der logischen Eigenschaften dieser Schnittstellen.

Ein Smartphone besteht aus sehr vielen Hardwarekomponenten. Jede einzelne Komponente wird benötigt, damit das Smartphone mit seinem kompletten Funktionsumfang funktioniert. Jedoch spielen einige Hardwarekomponenten eine besondere Rolle für NoRPG. Neben unverzichtbaren Komponenten wie den Prozessor, Speicher oder Akku zählen zu den Kernkomponenten der Touchscreen und der WLAN-Adapter. Der Touchscreen wird benötigt um die Eingaben des Spielers an das Spiel zu kommunizieren. Sei es die Steuerung des Charakters, das Ausfüllen des Registrierungsformulars oder das einfache betätigen eines Buttons. Ohne den Touchscreen können keine Eingaben ohne zusätzliche Peripherie an das Spiel kommuniziert werden. Der WLAN-Adapter ist zuständig für die Verbindung mit dem Internet. Ohne eine Internetverbindung ist nicht einmal der Login funktionsfähig bzw. es wäre nicht ohne Umstände möglich NoRPG aus dem Google Play Store herunterzuladen.

Obwohl es sich bei dem Server um einen virtuellen bzw. simulierten Server handelt, weiß NoRPG nicht, dass keine physische Hardware direkt benutzt wird. Für die App scheint es, als ob es mit einem physischen Server kommuniziert. Auch hier sind alle Komponenten des Server, auch wenn diese Simuliert sind, Teil der Hardware-schnittstelle.

## **Softwareschnittstellen**

Softwareschnittstellen spezifizieren die Schnittstellen mit anderen benötigten Softwareprodukten, welche die Nutzung derer Funktionalitäten ermöglicht.

Die vorinstallierte Software Google Play Store ist eine Plattform, die Musik, E-Books, Filme, Serien und insbesondere Apps anbietet. Der Google Play Store stellt eine Softwareschnittstelle zu NoRPG dar. NoRPG benötigt die Schnittstelle zur Spielplattform von Android, um die dort verfügbaren Lernspiel in NoRPG anzuzeigen bzw. als Download anzubieten.

Für die Verarbeitung der Toucheingaben gibt es ein C# Skript. Erst mit Hilfe dieser Software wird es möglich die Toucheingaben an das Spiel zu kommunizieren, damit die richtige Aktion in NoRPG ausgeführt wird.

Damit die lokalen Datenbanken der Clients und die Datenbank auf dem Server immer synchronisiert bleiben wird eine Datenbanksynchronisationssoftware benötigt, welcher die Synchronisation bei aktiver Internetverbindung übernimmt. Dabei handelt es sich um eine Softwareschnittstelle, da NoRPG die Funktionalität dieser Software verwendet.

Die letzte Softwareschnittstelle ist das Datenbank Management System (DBMS). Das DBMS ist die Verwaltungssoftware der Datenbank. Sie organisiert intern die strukturierte Speicherung der Daten und kontrolliert alle lesenden und schreibenden Zugriffe auf die Datenbank. Sie wird benötigt um den aktuellen Stand des Spieles zu speichern.

### 3.3.2 Funktionale Anforderungen

Use Cases dokumentieren Funktionalitäten eines Systems auf Basis von einfachen Modellen. In einem Use Case wird das nach außen sichtbare Verhalten eines Systems aus der Sicht des Nutzers beschrieben. Ein Nutzer kann hierbei eine Person, eine Rolle oder ein anderes System sein. Dieser Nutzer tritt als Akteur mit dem System in Interaktion, um ein bestimmtes Ziel zu erreichen.

Use Cases verwenden Activity Diagramme. Ein Activity Diagramm ist ein Verhaltensdiagramm der Unified Modeling Language (UML) und stellt die Vernetzung von elementaren Aktionen und deren Verbindungen mit Kontroll- und Datenflüssen grafisch dar.

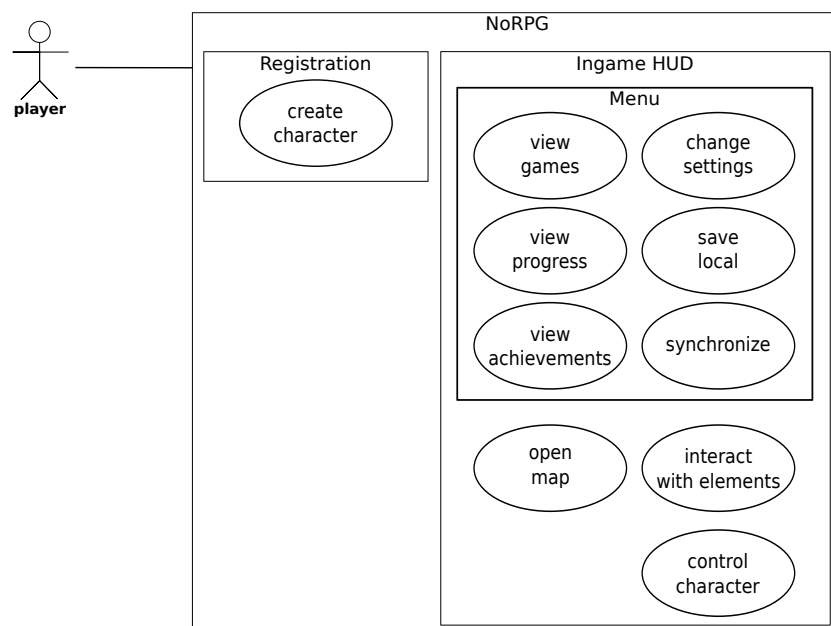


Abbildung 7: Overall Use Case Diagramm

Das abgebildete System in Abbildung 7 stellt die zu entwickelnde App für die User dar. Die App von NoRPG stellt die graphische Oberfläche und somit die beschriebenen Benutzerschnittstellen dar. Es sind nur die Funktionalitäten enthalten, die der Benutzer ausführen kann, also jene die über die Benutzerschnittstellen angesprochen werden können.

Use Cases wie Login oder Registrierung sind im Overall Use Case Diagramm nicht enthalten, da diese im Vergleich zu anderen Use Cases primitiv sind. Die abgebildeten Use Cases enthalten meistens mehrere Schritte bis die Aktion ausgeführt wird.

### Create character

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer seinen Charakter erstellen möchte. Dieser Use Case wird pro Account genau einmal im Registrierungsprozess ausgeführt.

Nach erfolgreicher Registrierung kann der Spieler seinen Charakter erstellen. Der User kann den Namen, das Geschlecht und das Aussehen des Charakters bestimmen. Ein möglicher Ablauf des Erstellungsprozesses kann aus Abbildung 8 entnommen werden.

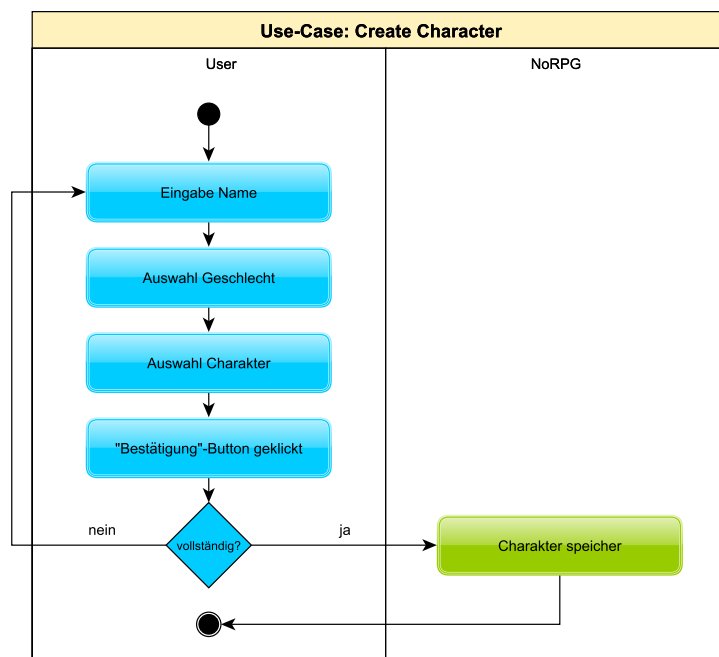


Abbildung 8: Activity Diagramm: Create Character

Der Ablauf des Prozesses kann allerdings variieren. Der Spieler kann beispielsweise erst das Geschlecht und das Aussehen bestimmen und anschließend seinem Charakter

einen Namen geben. Die Variationen können allerdings nur bei den Charaktereigenschaften auftauchen.

Bevor jedoch dieser Use Case ausgeführt werden kann, muss der Spieler das Registrierungsformular vollständig ausfüllen und erfolgreich abschließen. Zudem darf Account noch nicht existieren. Für den gesamten Registrierungsprozess ist eine aktive Internetverbindung notwendig, damit der neu angelegte Account mit den Spielereigenschaften direkt mit dem Server synchronisiert werden kann.

Nach der erfolgreichen Erstellung des Charakters, wird dieser in die Datenbank gespeichert und der User kann sich nun anmelden und Spiel starten. Sollte die Erstellung allerdings fehlschlagen, wird der User wieder auf den Hauptbildschirm von NoRPG weitergeleitet.

## Open map

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer die Karte der Spielwelt öffnet. Die Karte dient zur Orientierung und kennzeichnet den Aufenthalt des Spielers.

Der Anwender kann die Karte der aktuellen Spielwelt durch einen Klick auf die Mini-Map öffnen. Das Activity Diagramm in Abbildung 9) zeigt den vollständigen Ablauf.

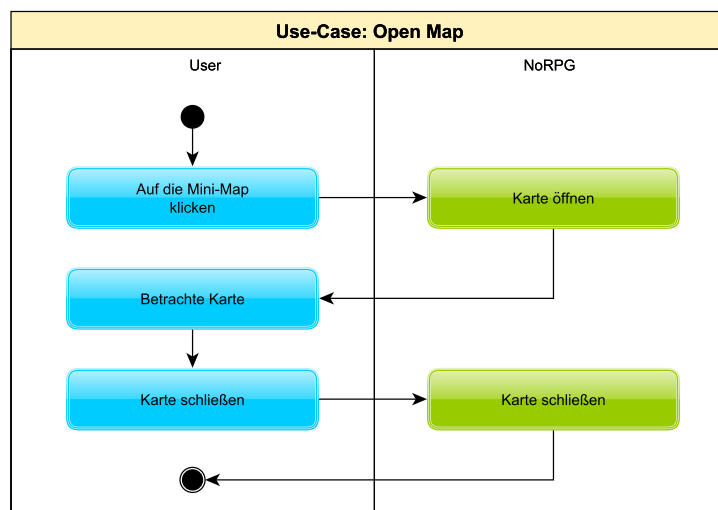


Abbildung 9: Activity Diagramm: Open Map

Die Vorbedingungen für diesen Use Case sind, dass der Spieler sich im Spiel befindet, das Menü geschlossen ist und der Spieler sich in keiner aktiven Non-Player Character (NPC) Interaktion befindet.

In einem Fehlerfall sollte sich die Karte schließen und der Benutzer wieder zum Spiel weitergeleitet werden.

## View games

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer die Liste der freigeschalteten und spielbaren Spiele öffnen will. Gefundene Spiele in NoRPG werden in diese Liste aufgenommen, so dass der Spieler an einem zentralen Ort alle spielbaren Lernspiele anschauen kann. Die Liste enthält den Namen des Spiels, einen Download-Link sowie die Zuordnung zum entsprechenden Standard. Dadurch ist es möglich auch ohne eine aktive Internetverbindung in NoRPG voranschreiten zu könne.

Das Activity Diagramm in Abbildung 10 hat im Vergleich zu den bisherigen betrachteten Activity Diagrammen eine Besonderheit. Die Aktivität "Google Play Store öffnen" findet außerhalb von NoRPG statt. Nach Beendigung diesen Schrittes wird der Spieler wieder zurück ins Spiel weitergeleitet. Allerdings wenn der Prozess von NoRPG währenddessen abgebrochen wird, startet das Spiel am letzten Speicherpunkt. Der Prozess beginnt indem der Spieler die Funktion im Menü aufruft.

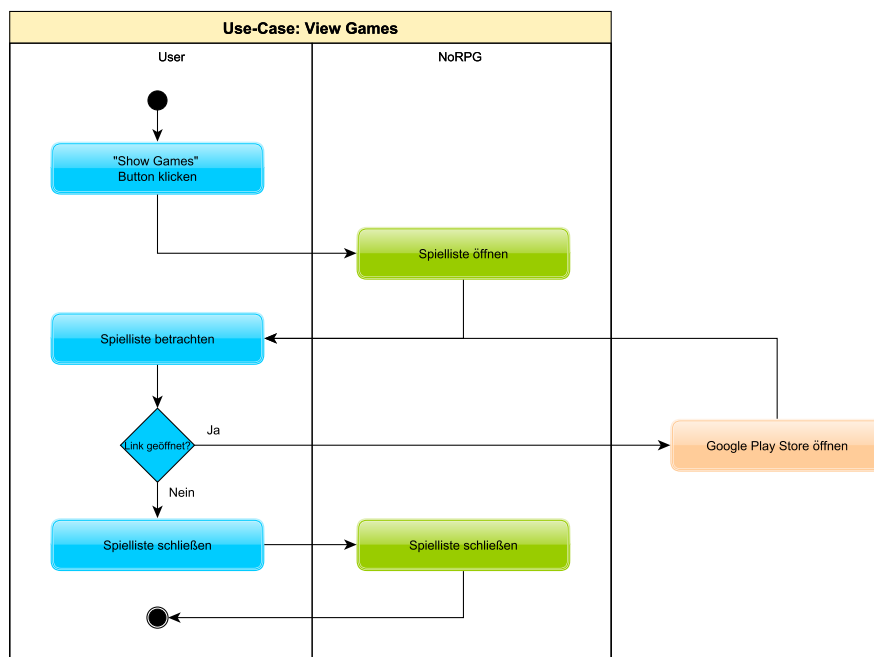


Abbildung 10: Activity Diagramm: Show Games

Der Benutzer muss sich im Spiel befinden und darf in keiner aktiven NPC Interaktion sein. Zudem muss das Menü vorher offen sein bevor dieser Use Case ausgeführt werden kann. Auch hier gilt im Fehlerfall, dass der Spieler wieder zurück ins Spiel kommt.

## View progress

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer seinen Fortschritt in NoRPG betrachten möchte. Dafür wird dem Spieler pro Schulfach jeweils ein Baumdiagramm präsentiert, der die Standards und deren Reihenfolge beinhaltet. Durch eine Kennzeichnung kann der Spieler sehen welche Standards abgeschlossen sind und fehlen.

Das Activity Diagramm in Abbildung 11 ähnelt dem vom Use Case "View Games". Die einzige Ausnahme dabei ist, dass dieses Diagramm keine Aktion hat, die außerhalb von NoRPG stattfindet. Die Fortschrittsanzeige wird im Menü geöffnet. Nachdem die Anzeige offen ist kann der Spieler zwischen den Standards wechseln.

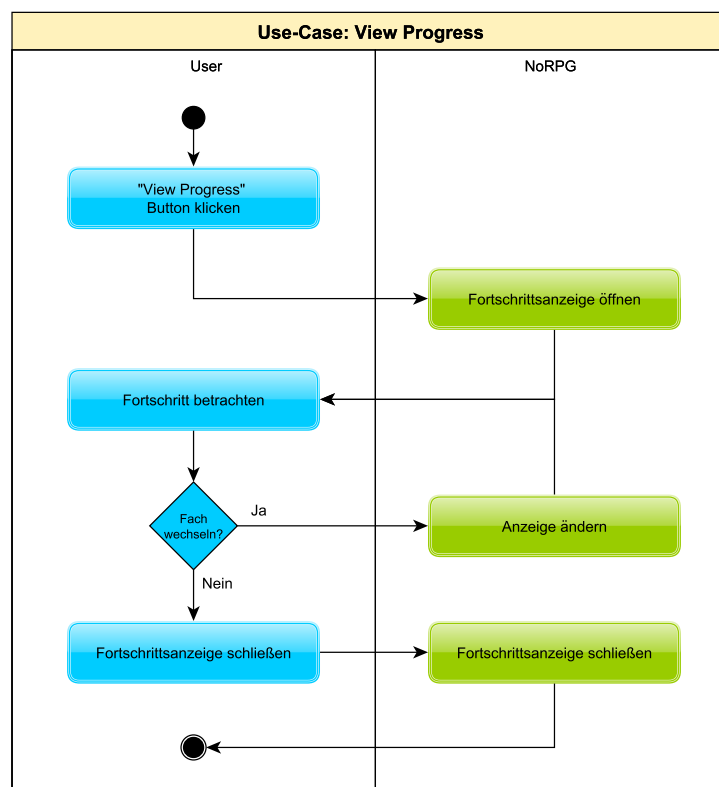


Abbildung 11: Activity Diagramm: View Progress

Dieser Use Case hat die gleichen Vorbedingungen wie der vorherige. Der Spieler muss sich im Spiel befinden, darf in keiner aktiven NPC Interaktion sein und das Menü ist vorher offen.

## View achievements

Dieser use Case beschreibt den Anwendungsfall, dass der Spieler seine Sammelgegenstände betrachten will. Neben den Standards kann der Anwender im Spiel Gegenstände finden, die ihn in der in Kapitel 2 beschriebenen Story weiterbringen. Diese Collectables sind auf den einzelnen Spielwelten verteilt und können dann, nachdem diese gefunden sind in der Ansicht zusammen mit den anderen betrachtet werden.

Im Prinzip ist dieses Activity Diagramm der gleiche wie der von Use Case "View progress". Der Spieler will seinen Fortschritt betrachten. Dieses mal allerdings seinen Fortschritt in NoRPG. Hier kann der Spieler zwischen den Welten wechseln anstatt zwischen den Fächern und es wird keine Baumstruktur präsentiert.

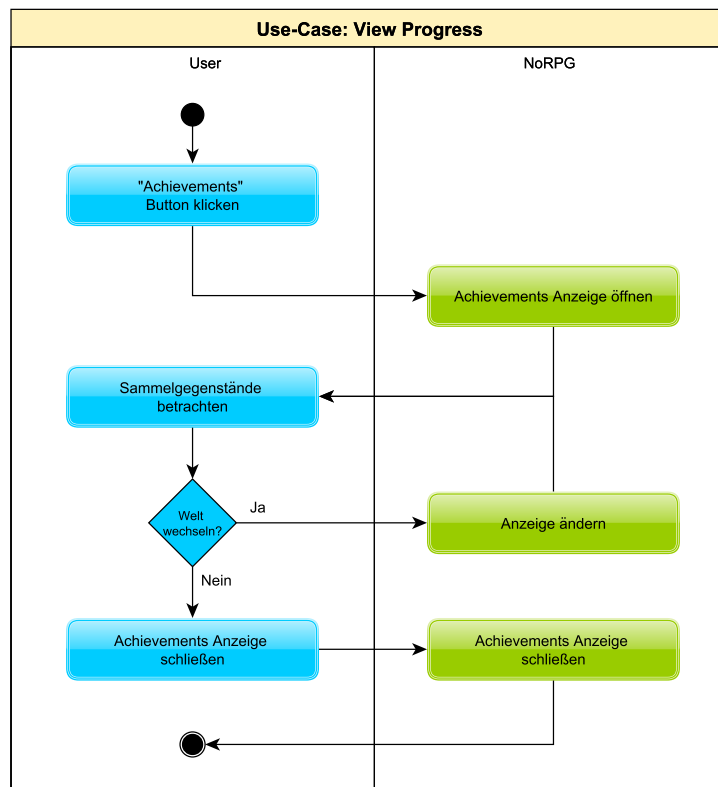


Abbildung 12: Activity Diagramm: Achievements

Dieser Use Case hat die gleichen Vorbedingungen wie der vorherige. Der Spieler muss sich im Spiel befinden, darf in keiner aktiven NPC Interaktion sein und das Menü ist vorher offen.



## **Change settings**

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer Einstellungen ändern möchte. In dem Rahmen dieser Studienarbeit wird es zwei Einstellungsmöglichkeiten geben. Der Spieler kann die Qualität des Spieles verändern. Ist die Option "High Quality" an, wird alles in höchster Qualität wiedergegeben. Ist jedoch diese Option aus, werden Hintergrundanimationen, die ressourcenaufwändig sind, nicht wiedergegeben. Diese Option ermöglicht es Smartphones mit schlechterer Hardwareausführung das Spiel ohne Probleme spielen zu können, bietet jedoch für neuere Smartphones die Möglichkeit NoRPG in vollen zu genießen.

Die zweite Einstellungsmöglichkeit ist, dass der Spieler die Audioausgaben des Spieles verändern kann. Zu einem Spielerlebnis gehört die Musik und die Soundeffekte. Diese kann der Spieler je nach Wünschen ein- oder ausschalten.

Die Spieloptionen können direkt im Menü durchgeführt werden. Es wird kein neues Fenster oder Ansicht dafür geöffnet. Der Vollständigkeit halber ist der Ereignisablauf im Anhang in Abbildung ?? zu sehen.

Die Vorbedingung zur Änderung der Spieleinstellungen sind, dass der Anwender sich im Spiel und sich in keiner aktiven NPC Interaktion befindet. Bei erfolgreicher Änderung sollten diese Einstellungen für diesen Account gespeichert werden, damit diese bei jedem Login, auch auf verschiedene Geräte, übernommen werden. Im Fehlerfall werden die Default-Optionen angewendet.

## **Save local**

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer seinen aktuellen Fortschritt manuell speichern möchte. NoRPG speichert bei bestimmten Ereignissen automatisch. Beispielsweise wenn der Spieler in eine andere Welt reist oder eine Truhe findet. Allerdings kann der Spieler seinen aktuellen Stand speichern. Es werden neben Fortschritt, Erfolge oder Charaktereigenschaften noch die Spieleinstellungen und sogar der aktuelle Standort des Spielers gespeichert.

Der Spieler kann diese Funktion direkt im Menü durchführen. Auch hier befindet der Vollständigkeit halber das Activity Diagramm im Anhang in Abbildung ??.

Wenn das Speichern erfolgreich war, wird der Spieler durch eine kurze Nachricht informiert. Wenn keine Information erscheint kann davon ausgegangen werden, dass der Vorgang fehlgeschlagen ist. Im Fehlerfall muss die Aktion erneut durchgeführt werden.

## Synchronize

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer seinen lokalen Speicherstand mit dem Server manuell synchronisieren möchte. NoRPG synchronisiert automatisch, wenn eine aktive Internetverbindung vorhanden ist beim Einloggen und vor dem Ausloggen. Die Synchronisation wird nicht so oft wie das Speichern ausgeführt, da der Vorgang länger dauert. Ebenfalls wird die Synchronisation nicht so häufig benötigt, da der Server als eine Art Back-Up verwendet wird. Also für den Fall, dass der Spieler sich mit einem anderen Gerät anmelden möchte.

Der Spieler kann diese Funktion direkt im Menü durchführen. Auch hier befindet der Vollständigkeit halber das Activity Diagramm im Anhang in Abbildung ??.

Wenn das Synchronisieren erfolgreich war, wird der Spieler durch eine kurze Nachricht informiert. Wenn keine Information erscheint kann davon ausgegangen werden, dass der Vorgang fehlgeschlagen ist. Eine häufige Ursache für den Fehlschlag kann die fehlende Internetverbindung sein.

## Control character

Dieser Use Case beschreibt den Anwendungsfall, dass der Spieler seinen Charakter in NoRPG durch die Spielwelt kontrollieren möchte. Dafür verwendet der Spieler das Steuerkreuz, welches sich in der linken unteren Ecke des HUD befindet (vgl. Abbildung 4).

Das Activity Diagramm dieses Use Cases in Abbildung 13 ist im Vergleich zu den bisherigen Diagrammen besonders, denn es existiert keinen Endpunkt.

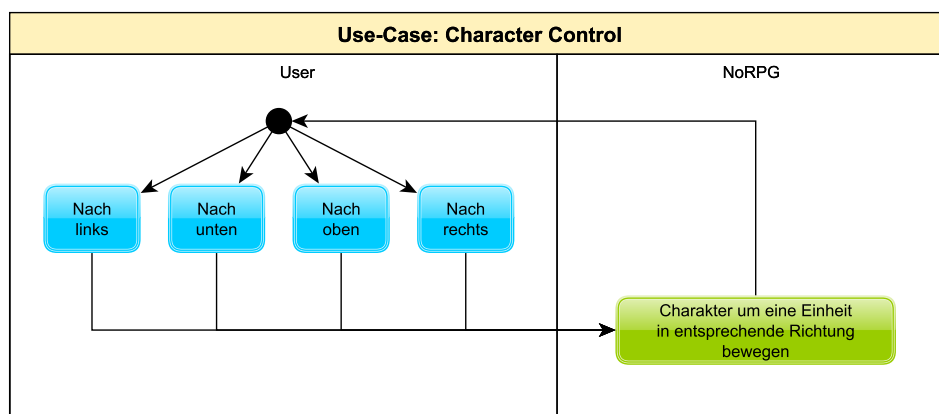


Abbildung 13: Activity Diagramm: Character Control

Wenn der Spieler das Steuerkreuz nach oben bewegt, wird der Charakter des Spielers um eine Einheit nach vorne bewegt. Nach dem Abschluss dieser Aktion kann der Spieler den Charakter direkt weiter bewegen. Wenn der Touchscreen in eine Richtung gehalten wird, dann wird dieser Ablauf ganze Zeit durchgeführt, bis die Touchscreen losgelassen wird.

Die Vorbedingung für diesen Use Case ist, dass der Spieler sich im Spiel befindet und das Menü geschlossen ist.

## **Interact with elements**

Dieser Use Case beschreibt den Anwendungsfall, dass der Benutzer mit einem Element im Spiel interagieren möchte. Dafür hat der Spieler 2 Tasten auf dem HUD (vgl. Abbildung 4), die dem Spieler die Möglichkeit gibt die Interaktion zu starten oder zu beenden.

In NoRPG wird es verschiedene Elemente geben, mit denen der Spieler interagieren kann.

- **NPCs:** Der Spieler kann sich mit unterschiedlichsten NPCs unterhalten, die überall in allen Spielen vorkommen. Die NPCs helfen dem Spieler, indem diese Tipps und Hinweise in den Unterhaltungen nennen.
- **Truhen:** Der Spieler kann mit Truhen interagieren. Die Truhen sind in den einzelnen Welten verteilt und enthalten die Sammelgegenstände. Durch die Interaktion öffnet sich die Truhe und der Sammelgegenstand erscheint. Nachdem die Truhe geöffnet wurde bleibt die Truhe offen und es kann nicht weiter mit ihr interagiert werden. Es handelt sich um eine einmalige Aktion.
- **Spielehändler:** Spielehändler tauchen überall auf. Diese bieten die verschiedenen Lernspiele für Standards an. Die Spielehändler können menschliche Händler oder andere Formen annehmen um die verschiedene Welten zu repräsentieren. Durch die Interaktion startet der Spieler eine Unterhaltung in dem der Spieler Informationen über den Standard erhält und das Spiel freischaltet.
- Die Interaktion mit der Umwelt (Bäume, Tiere, Gebäude) starten zunächst keine Aktion.

Eine Interaktion kann nur dem "A"Button gestartet werden. Wenn ein Element zur Interaktion vorhanden ist startet erst die Interaktion. Anschließend kann der Spieler mit dem "B"Button reagieren. Der "B"Button ist zum Abbrechen der Interaktion und beendet diesen Prozess bzw. überspringt ihn. Zum Beispiel in der Interaktion mit dem

Händler sorgt der "B" Button dafür, dass die Unterhaltung übersprungen wird und direkt die Spiele angezeigt werden. Mit dem "A" Button wird die Interaktion fortgeführt bis keine Aktion mehr vorhanden ist, also in unserem Händlerbeispiel keine Unterhaltung mehr gibt.

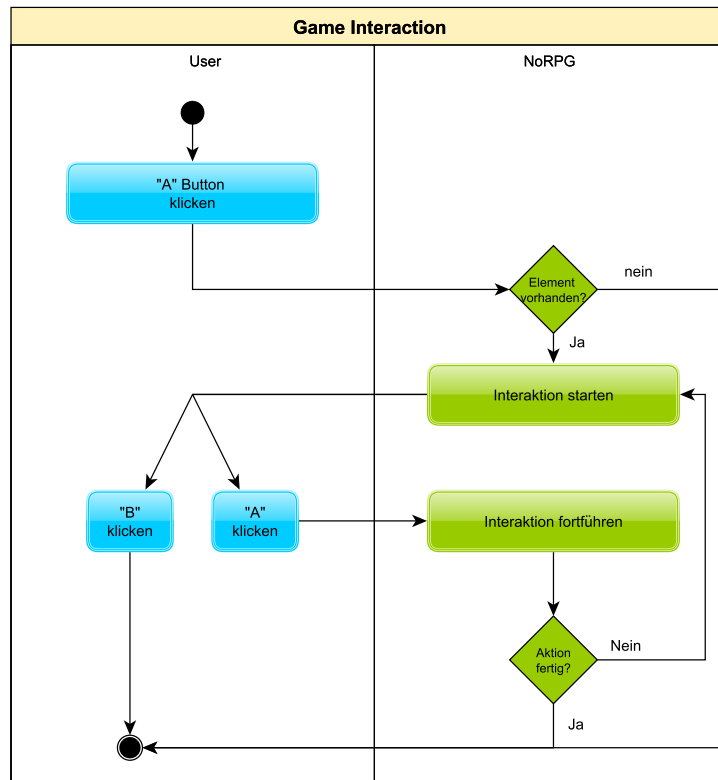


Abbildung 14: Activity Diagramm: Game Interaction

Die Vorbedingungen für diesen Use Case sind, dass der Spieler sich im Spiel befindet und das Menü sowie die Karte geschlossen sind. Je nach Element wird ein anderes Ergebnis erwartet. Beispielsweise wird bei einer Interaktion mit der Truhe das Ergebnis erwartet, dass die Truhe offen bleibt, der Sammelgegenstand in die Liste übernommen wurde und der Spielstand automatisch lokal gespeichert wird.

### 3.3.3 Performanz Anforderungen

Dieser Abschnitt des SRS legt sowohl die statischen als auch die dynamischen numerischen Anforderungen an die Software.

Ein sehr wichtige Anforderung betrifft die Antwortzeit des Systems. Diese Anforderung ist besonders wichtig, da der Nutzer die direkten Auswirkungen in der App mitbekommt. Eine Antwortzeit von 0,1 Sekunden gibt dem Benutzer das Gefühl, dass das System sofort reagiert. Bei einer Sekunden denkt der Benutzer immer noch, dass die

Aktion ununterbrochen durchgeführt wird, jedoch bemerkt der Benutzer ein Verzögern. Im Worst-Case liegt die Grenze für die Aufmerksamkeit des Benutzers bei zehn Sekunden<sup>8</sup>.

Für die Performanz Anforderungen wird zwischen der App und dem Server unterschieden. 95% der Transaktionen in der App sollten in weniger als einer Sekunde verarbeitet werden. Aufwändigere Prozesse wie die Synchronisation oder das wechseln der Spielwelten dürfen minimal länger brauchen. Für den Server gilt, dass 80% der Transaktionen in weniger als einer Sekunde verarbeitet werden. Es sind nur 80% notwendig, da der Server nur für die Synchronisation und Speicherung verwendet wird. Der prozentuale Anteil der Synchronisation auf dem Server ist wesentlich größer als die in der App.

Der Workload ist die Anzahl an Verarbeitungen, die das System in einer gegebenen Zeit durchführen kann. Die App selbst hat keine hohe Arbeitsbelastung, da diese nur einen Benutzer gleichzeitig zulässt und von der Hardware des Spielers abhängt. Vielmehr gilt es zu betrachten, mit wie viel Belastung der Server umgehen kann. Der Anzahl an Spielern ist mehr oder weniger keine Grenze gesetzt, da für jeden einzelnen Spieler zunächst eine Zeile in der Datenbank hinterlegt wird. Die Anzahl an gleichzeitigen Benutzern, die mit dem Server kommunizieren wollen, ist allerdings begrenzt. Da der Server auch sein Hardwarebegrenzungen hat kann es nur eine bestimmte Anzahl an gleichen Synchronisationsanfragen entgegennehmen. Wenn diese Anzahl übersteigt wird arbeitet der Server nach dem First In First Out (FIFO) Prinzip. Die Clients, die als erste eine Synchronisationsanfrage senden, werden auch als erstes bearbeitet. Als eine Art der Erweiterung spezifiziert die Lastskalierbarkeit die Menge an Daten, die innerhalb bestimmter Zeitperioden sowohl bei normaler als auch bei maximaler Belastung verarbeitet werden sollen.

### 3.3.4 Datenbank Anforderungen

Die beiden eingesetzten Datenbanken unterscheiden sich von ihren Anforderungen.

Bei der Datenbank in der App handelt es sich um eine embedded, zu deutsch eingebettete, Datenbank. Sie wird zusammen mit der App ausgeliefert. Die Datenbank enthält zunächst die Spieldaten. Das ist eine Liste der Standards mit den dazugehörigen Spielen. Neben diesen Daten werden lokal noch die Daten des Spielers gespeichert. Die Spielerdaten beinhalten gespeicherte Einstellungen, die Charaktereigenschaften und den Fortschritt. Dadurch wird das Spielen ohne eine aktive Internetverbindung ermöglicht.

---

<sup>8</sup> vgl. Andrew Lee [performance1]

Die embedded Datenbank wird oft verändert, da bei jedem Fortschritt automatisch gespeichert wird. Zudem werden die Daten der Datenbank beim Einloggen und vor dem Ausloggen immer mit dem Server synchronisiert. Die Daten werden beim Ausloggen gelöscht, damit sich verschiedene Spieler auf einem Smartphone anmelden können und die Datensicherheit gewährleistet wird.

Die Integrität der Datenbank muss immer gewährleistet werden, damit die Benutzer den eigenen Fortschritt nicht fälschen können. Dies kann mit Prüfsummen und Versionierung gewährleistet werden.

Die Datenbank auf dem Server enthält alle Spielerdaten und Spieldaten. Bei den Spieldaten handelt es sich um die gleichen Daten, die auf den lokalen Datenbanken gespeichert sind. Allerdings sind auf dem Server alle Spielerdaten gespeichert, damit die Benutzer von unterschiedlichen Geräten spielen können.

Diese Datenbank wird nicht oft verändert bzw. aktualisiert, da die Datenbank nur bei der Synchronisation verändert wird oder wenn neue Standards oder Spiele eingefügt werden. Das hat den Vorteil das nicht sehr viele Anfragen an den Server gestellt werden, wodurch der Workload gering gehalten wird. Zugriffen kann der Spieler nur über die App. Administratoren haben die Möglichkeit den Server über eine Virtual Machine zu erreichen und Änderungen durchzuführen.

Auch hier muss die Integrität der Datenbank immer gewährleistet werden.

### **3.3.5 Entwurfsbeschränkungen**

Bei der Entwicklung der App gilt es zu beachten, dass die Hardware-Anforderungen nicht zu hoch sind. Als Maßstab für die Hardware wird das Smartphone Samsung Galaxy S4 genommen. Grund dafür ist das Preis-Leistungsverhältnis des S4 und die Tatsache, dass Samsung eine sehr verbreitete Marke ist.

Das Samsung Galaxy S4 kostet auf Amazon (Stand 27.01.2017) ungefähr 225 Euro. Das S4 hat vorinstalliert Android 4.2 Jelly Bean und die Samsung Touchwiz 4.0 Oberfläche. Ein Update auf Android 5.0 Lollipop ist allerdings möglich. Das S4 hat 2 GByte Arbeitsspeicher und 16 GByte internen Speicher, welcher durch eine Speicherkarte erweitert werden kann. Bei dem Prozessor handelt es sich um einen Qualcomm Snapdragon 600 mit vier Kernen, einer Taktfrequenz von 1,9 GHz und einer 32-bit Architektur<sup>9</sup>. Für eine Internetverbindung bietet Samsungs Galaxy S4 eine drahtlose Schnittstelle, die nach IEEE 802.11a/ac/b/g/n funktioniert.

---

<sup>9</sup> vgl. Google [s4Google]

Speicherbedarf darf nicht größer als 200MB sein, geplant sind allerdings 150MB. Beim RAM dürfen nicht mehr als 1GB sein, geplant sind allerdings 500MB.

### 3.3.6 Benutzerfreundlichkeit

Das Ziel der Benutzerfreundlichkeit ist eine hohe Ergonomie. Die Software-Ergonomie bezeichnet die Anpassung an die kognitiven und physischen Fähigkeiten bzw. Eigenschaften des Benutzers, also seine Möglichkeiten zur Verarbeitung von komplexen Informationen aber auch die Anpassung softwaregesteuerter Merkmale der Darstellung, wie Farben und Schriftgröße.

Damit die Benutzeroberfläche freundlich für den Benutzer ist, muss sie in das Profil des Benutzers passen. Da NoRPG sich grundsätzlich an Kinder richtet muss die Bedienung und Gestaltung der App kindgerecht sein. Damit eine App als kindgerecht bezeichnet werden kann muss es nach Wendy B. von Intel vier Prinzipien erfüllen<sup>10</sup>.

1. Freiheit: Die Fähigkeit, sich in der App innerhalb einer kontrollierten Umgebung frei bewegen zu können. Dieses Prinzip verfolgen auch Rollenspiele. Durch die offene Spielwelt kann der Spieler selbst entscheiden wohin er als nächstes hin möchte. Allerdings gilt es bei Kindern diese Umgebung zu kontrollieren, indem bestimmte Bereiche festgelegt werden müssen.
2. Komfort: Die App sollte stimulieren, jedoch darf es nicht übertrieben werden. Dies wird als Balanceakt bezeichnet. Abwechslungsreiche Stimulationen sind definitiv notwendig, beispielsweise durch die Animationen, Farben oder Musik. Dadurch wird die Wahrnehmung und Aufmerksamkeit erhöht. Allerdings ist die Linie zwischen Stimulation und Lärm bzw. zu viel Stimulation sehr dünn und kann leicht überschritten werden.
3. Vertrauen: Kinder, genau wie Erwachsene, müssen sich kompetent fühlen und wollen ihren Aktionen vertrauen.
4. Kontrolle: Kinder wollen das Gefühl, dass sie etwas vollbringen wenn sie in der App interagieren. Es werden Ziele gesetzt und Entscheidungen getroffen.

Für die kindgerechte Gestaltung gibt es allerdings noch weitere Kriterien. Es dürfen keine In-App-Käufe angeboten, keine Werbung platziert oder zu Social Media oder ähnlichen Seiten verlinkt werden. Diese Gegenstände haben in einem kindgerechten Spiel nichts zu tun. Für die Installation gilt es zu beachten, dass so wenig Berechtigungen wie notwendig verlangt werden<sup>11</sup>.

---

<sup>10</sup> vgl. Wendy B. [intelKids]

<sup>11</sup> vgl. klick-tipps.net [appsforkids]

Es gilt, desto einfacher die App gestaltet ist, desto mehr Kinder verstehen diese. Ein Beispiel dafür ist die Charaktersteuerung. Moderne Spiele haben keine sichtbaren Steuerkreuze, der Spieler muss nur eine Ziehbewegung in die Richtung machen, in der er sich bewegen möchte. In NoRPG allerdings wird ein sichtbares Steuerkreuz gewählt, wie dies schon von vielen Geräten wie vom klassischen Gameboy oder von Playstation verwendet wird. Navigation soll schnell erkennbar und nachvollziehbar sein.

Schließlich gilt es die erwachsenen Nutzer nicht zu vergessen! Gemeint sind Eltern, Erziehungsberechtigte, Lehrer und jeder, der mit der App interagieren kann. Erwachsene spielen genauso eine wichtige Rolle. Egal ob diese die Kinder unterstützen, überwachen oder selbst spielen. Es ist wichtig, dass sich Eltern an die Entwickler melden können, wenn sie etwas nicht kindgerecht halten oder wenn es Probleme gibt<sup>12</sup>.

### **3.3.7 Zuverlässigkeit**

Alle implementierten Produktfunktionen sollten zur Auslieferung korrekt und zuverlässig funktionieren. Dazu zählt auch, dass die Funktionen in vertretbaren Zeiten terminieren. Beispielsweise sollte die Anmeldung funktionieren, wenn der Spieler registriert ist und die richtigen Benutzerdaten eingegeben hat, oder die Benutzereingaben für die Charaktersteuerung sollen korrekt interpretiert werden.

Eine besondere Wichtigkeit hat die Implementierung der in Kapitel 2 beschriebenen Common Core State Standards. Diese sind wichtig für die Reihenfolge der spielbaren Lernspiele, damit ein Spieler mit einem Skilllevel der ersten Klasse in Geometrie keine Lernspiele für die fünfte Klasse angezeigt kriegt. Erst dadurch wird gewährleistet und kann sichergestellt werden, dass der Spieler die Lerninhalte korrekt vermittelt kriegt.

### **3.3.8 Verfügbarkeit**

Da bei jeder App eine lokale Datenbank vorhanden ist, muss der Server nicht die ganze Zeit verfügbar sein. Das gilt jedoch nur für die Benutzer, die NoRPG schon heruntergeladen und sich angemeldet haben. Wenn diese Bedingungen erfüllt sind, wird der ganze Fortschritt lokal gespeichert und kann mit dem Server manuell oder automatisch synchronisiert werden. Für den Fall, dass der Benutzer sich von einem anderen Gerät anmelden möchte, muss der Server verfügbar sein.

Daher werden nur die Verfügbarkeit des Logins bzw. Registrierung und der Synchronisation bewertet. Der Login muss eine Verfügbarkeit von 99% haben, wohingegen bei der Synchronisation eine Verfügbarkeit von 90% ausreicht.

---

<sup>12</sup> vgl. Becky White [smashMagazin]



### **3.3.9 Sicherheit**

Bei Sicherheit wird zwischen zwei unterschiedlichen Typen unterschieden: Security und Safety. Security ist der Schutz vor absichtlichen Bedrohungen, wenn ein Angreifer absichtlich das Systems angreift. Im Gegensatz dazu ist Safety der Schutz vor unbeabsichtigten Bedrohungen, wenn der Benutzer durch Zufall die Sicherheitsmechanismen umgeht indem er etwas nicht beabsichtigtes ausführt.

Die Kommunikation mit dem Server und mit der lokalen eingebetteten Datenbank müssen verschlüsselt werden, damit die Credentials bei den Anmeldung oder bei der Registrierung nicht mitgelesen werden können. Des Weiteren müssen die Daten auf der lokalen Datenbank validiert werden, bevor der Server synchronisiert wird, denn es wird unter anderem auch der Spielfortschritt der Benutzer synchronisiert. Die Veränderung des Spielfortschritts wird als Schummeln bzw. Cheaten behandelt.

Die Anmeldung bzw. Registrierung ist notwendig, um NoRPG spielen zu können. Deswegen müssen die Accounts der Benutzer verschlüsselt gespeichert werden und die Passwörter dürfen bei der Anmeldung nur mittels One-Way-Functions verglichen werden. Nicht registrierte bzw. unautorisierte Benutzer erlangen keinen Zugriff auf das System.

Die gespeicherte Daten dürfen an andere Tools nur anonymisiert weitergegeben werden, für beispielsweise Analysezwecke. Diese Kommunikation mit anderen System oder Applikationen darf nur verschlüsselt geschehen.

### **3.3.10 Wartbarkeit**

Der Code von NoRPG sollte so geschrieben werden, dass der Code die Umsetzung neuer Funktionen begünstigt. Deshalb sollte die Komplexität des Quellcodes so gering wie möglich gehalten werden, indem entsprechende Methoden wie das Model-View-Controller (MVC) Pattern umgesetzt werden. Des Weiteren sollte das System von NoRPG Schnittstellen jeglicher Art anbieten, um das System durch weitere Komponenten wie ein Web-Tool für Administratoren zu erweitern.

Neben der Erweiterbarkeit sollte NoRPG für Fehlerfälle eine Testumgebung anbieten, um das Testen der Anwendung auf unterschiedliche Funktionen zu ermöglichen und gegebenenfalls Fehler wiederholen und simulieren zu können.

### 3.3.11 Portabilität

Die App NoRPG ist zunächst nur für Android geplant. Andere Betriebssysteme, wie Windows Phone von Microsoft oder iOS von Apple, sind vorerst nicht vorgesehen.

Bei der vorhandenen Breite an Varianten von Android-Smartphones ist sehr wichtig, dass die Portabilität innerhalb von Android Smartphones gewährleistet wird. Neben bekannten Smartphoneherstellern wie Samsung, LG oder HTC gibt es zahlreiche weitere Hersteller die auf das Android Betriebssystem setzen. Jeder Hersteller hat dabei eine große Palette an Smartphone-Modellen, wie bei Samsung die Samsung Galaxy S-Reihe, welches aktuell in der siebten Generation erhältlich ist<sup>13</sup>, oder die Samsung Galaxy Note-Reihe. Die App NoRPG muss auf allen Android-Smartphones funktionieren, solange diese die Mindestanforderungen an Software und Hardware erfüllen. Dabei muss sich die App beispielsweise an die Auflösung des Smartphones anpassen.

Die Portabilität beschreibt nicht nur die technische Sicht sondern auch in welchen Ländern und in welchen Sprachen NoRPG verfügbar sein wird. Der Release findet in allen Ländern statt, in denen der Google Play Store verfügbar ist. Zunächst wird NoRPG nur in Englisch verfügbar sein.

---

<sup>13</sup> vgl. Philippe Fischer, Michael Huch [[computerBildSamsung](#)]

## 4 Technische Grundlagen

Nachstehend werden die technischen Grundlagen erläutert und es wird kurz auf diese eingegangen. Dabei wird mit den Entwicklungsumgebungen begonnen.

### 4.1 Unity

Bei Unity handelt es sich um eine Entwicklungs- und Laufzeitumgebung, mit deren Hilfe graphisch aufwändige Projekte umgesetzt werden können. Dazu gehören unter anderem Videospiele, aber auch Lernprogramme oder Apps können damit umgesetzt werden. Dazu können in Unity 3D-Projekte, aber auch 2D beziehungsweise 2,5D Projekte umgesetzt werden (Dabei handelt es sich um eine Mischung von 2D und 3D Spielen). Dank Unity sind diese Projekte nach der Entwicklung plattformübergreifend einsetzbar.<sup>14</sup> Die Entwicklungsumgebung teilt sich dabei in verschiedene Bereiche auf. Diese sind dabei an einen 3D-Editor angelegt.

Das „Scene“ Fenster ist in den Standardeinstellungen in der Mitte des Bildschirms zu sehen (Siehe Abbildung 15)<sup>15</sup>.

Hier wird immer die aktuelle Szene dargestellt. Darüber hinaus kann hier mit Objekten der Szene interagiert werden, um diese zu verändern, zum Beispiel an eine andere Stelle platzieren oder zu skalieren.

Wird ein Objekt in diesem Fenster ausgewählt, befinden sich im Bereich „Inspector“ zusätzliche Einstellmöglichkeiten<sup>16</sup>. Diese variieren je nach gewähltem Objekt. Auch hier können die Position oder die Skalierung eines Objektes verändert werden, allerdings werden hier auch darüber hinausgehende Eigenschaften der Objekte verändert. Dazu zählen die visuellen sowie die physischen Eigenschaften der Objekte.

Ein weiterer Weg, ein Objekt auszuwählen, ist das markieren im „Hierarchy“ Fenster<sup>17</sup>. In diesem Fenster werden alle Objekte der aktuellen Szene aufgelistet. Dabei wird zwi-

---

<sup>14</sup> vgl. Unity3D [unity1] (2017)

<sup>15</sup> vgl. Unity3D [unity2] (2017)

<sup>16</sup> vgl. Unity3D [unity3] (2017)

<sup>17</sup> vgl. Unity3D [unity4] (2017)

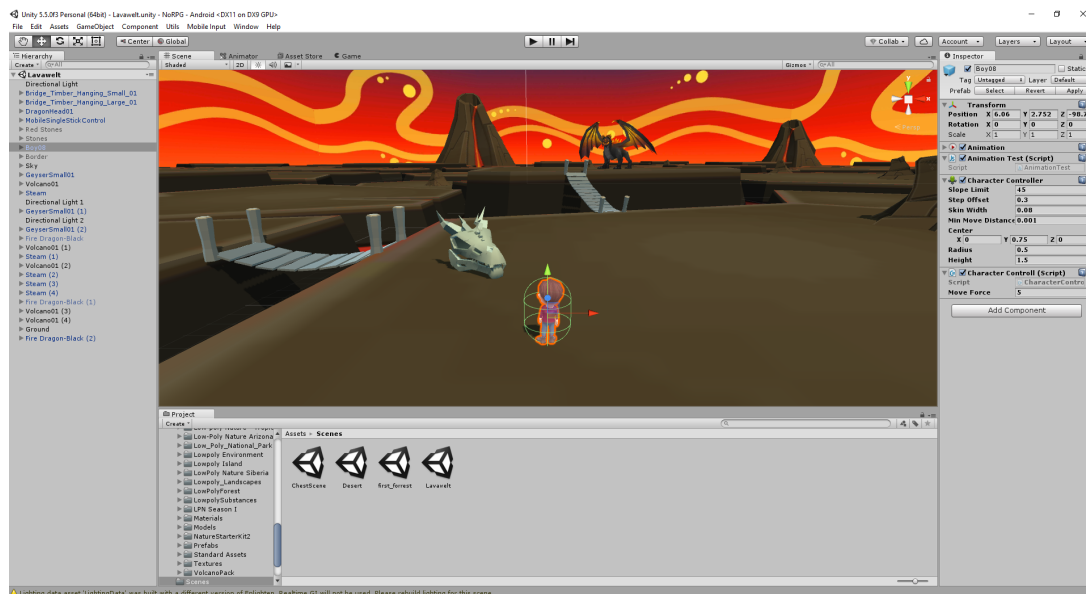


Abbildung 15: Darstellung von Tablemappings

schen unterschiedlichen Ebenen unterschieden, so dass genau gesehen werden kann welche Objekte zusammengehören.

Um alle Dateien zu sehen, die zu einem Projekt gehören, gibt es das „Project“ Fenster<sup>18</sup>. Die Dateien werden dabei nach der vorliegenden Ordnerstruktur angezeigt. In diesem Fenster können Ordner, sowie andere Dateien erstellt werden.

Ein weiteres wichtiges Fenster ist das „Game“ Fenster<sup>19</sup>. Hier kann das fertige Projekt angesehen werden. Dazu gibt es oben in der Mitte der Benutzeroberfläche, ein Start, Pause und Vorlauf Button. Mit Hilfe derer kann das Programm, bevor es auf der Zielplattform abgespielt wird, in der Entwicklungsumgebung gerendert werden. Der Code wird von Unity JustIn-Time (JIT) kompiliert, und anschließend auf Mono oder dem Microsoft .NET Framework ausgeführt. Der Code steht in sogenannten Skripten, die in C#, UnitySkript (ähnlich JavaScript) oder Boo geschrieben sind.

Wenn während der Laufzeit oder im Vorfeld beim Kompilieren ein Fehler auftritt, wird dieser im „Console“ Fenster ausgegeben. Darüber hinaus werden hier Meldungen angezeigt, die explizit in den Skripten programmiert wurden. Damit es zu keinen Fehlern kommt, gibt es in Unity Tests, die eine Szene auf Korrektheit prüft. Die sogenannten Integrationstest simulieren eine Szene, damit verschiedene Objekte auf ihre Eigenschaften geprüft werden können.

Nachdem nun die Benutzeroberfläche ausführlich erklärt wurde, wird nun auf die Begriffe Prefabs und Skripte eingegangen, da diese Essentiell für den Umgang mit Unity

<sup>18</sup> vgl. Unity3D [unity5] (2017)

<sup>19</sup> vgl. Unity3D [unity6] (2017)

sind.

### **Prefabs**

Bei Prefabs handelt es sich um fertige Objekte, die in Szenen verwendet werden können<sup>20</sup>. Diese können dabei als Vorlage gesehen werden, damit nicht jedes Objekt mit gleichen Eigenschaften erneut erstellt werden müssen.

### **Skripte**

In Skripten befindet sich die Logik von Objekten<sup>21</sup>. Ein Beispiel dafür ist das Öffnen einer Truhe. Wenn der Benutzer die Truhe anklickt und öffnen will, steht in einem Skript, was die Truhe zu tun hat. In diesem Beispiel also, das sie sich öffnen soll.

Diese Logik kann mithilfe einer Entwicklungsumgebung angepasst werden. Die dazu geeignete Entwicklungsumgebung wird bei der Installation von Unity mitgeliefert. Dabei handelt es sich um Microsoft Visual Studio.

## **4.2 Visual Studio**

Mit Microsoft Visual Studio ist es möglich in verschiedenen Programmiersprachen zu programmieren<sup>22</sup>. Bei der Installation von Unity wird dieses dabei zusätzlich installiert. Dadurch können die Skripte aus Unity in Visual Studio geöffnet und bearbeitet werden. Zusätzlich zu Visual Studio werden auch verschiedene Plug-Ins für die IDE installiert. Dabei handelt es sich unter anderem um eine ausführliche Dokumentation von allen in Unity zur Verfügung stehenden Methoden und Klassen sowie um Testtools, um verschiedene Tests auszuführen.

## **4.3 C Sharp**

C# (gesprochen C Sharp) ist eine Programmiersprache welche von Microsoft entwickelt wurde. Sie wurde zusammen mit „.NET 1.0“ 2002 in der Version 1 veröffentlicht und ist mittlerweile in Version 6 verfügbar.<sup>23</sup>

C# orientiert sich dabei an den Programmiersprachen C, C++, Java, Delphi und Haskell und nutzt deren grundlegenden Konzepte. Aufgrund der Ähnlichkeit zu diesen Sprachen handelt es sich bei C# ebenfalls um eine Objektorientierte Sprache.

---

<sup>20</sup> vgl. Unity3D [**unity7**] (2017)

<sup>21</sup> vgl. Unity3D [**unity8**] (2017)

<sup>22</sup> vgl. Microsoft [**microsoft1**] (2015)

<sup>23</sup> vgl. Microsoft [**microsoft2**] (2015)

Nachfolgend wird nun auf die Grundlegenden Konventionen eingegangen, denen C# zugrunde liegen. Danach wird auf die Verwendung in Unity Skripten eingegangen.

### 4.3.1 Allgemeiner aufbau C#

Der allgemeine Aufbau von C# wird hier am Beispiel von einem Hello World Programm in Listing 1 dargestellt.

Dabei ist in Zeile eins ein Text zu sehen. Vor diesem stehen zwei Slashes. Damit wird der Text als Kommentar gekennzeichnet. Daruch wird dieser Abschnitt vom Compiler beim Compilieren ignoriert. In Zeile zwei wird das Schlüsselwort „using“ gefolgt von einem Namen, in diesem Fall „System“, genutzt. Dieses dient dazu um das Package System in dem Programm zu nutzen.

```
1 // A Hello World! program in C#.
2 using System;
3 namespace HelloWorld
4 {
5     class Hello
6     {
7         static void Main()
8         {
9             Console.WriteLine("Hello World!");
10
11             // Keep the console window open in debug mode.
12             Console.WriteLine("Press any key to exit.");
13             Console.ReadKey();
14         }
15     }
16 }
```

Listing 1: Hello World in C#

Als nächstes wird ein Namespace definiert. Innerhalb von diesem eine Klasse namens „Hello“. Innerhalb von dieser wiederum befindet sich der auszuführende Code.

Dieser steht in einer Methode, die in Zeile sieben definiert wird. Bei dieser Methode handelt es sich um die Main() Methode. Diese wird beim Programmstart immer zuerst ausgeführt. In dieser Methode gibt es vier Zeilen Code, darunter ein Kommentar(in Zeile elf). Bei den anderen drei Zeilen handelt es sich um Konsolenausgaben bzw. Konsoleneingaben. Die Zeilen neun und zwölf geben jeweils Text auf der Konsole aus. In Zeile 13 wird dagegen ein Zeichen eingelesen, welches vom Benutzer eingegeben wird.

### 4.3.2 Unity Skripte

Die in Unity verwendeten C# Skripte erben standardmäßig von der Klasse `MonoBehaviour` wie in Listing 2 zu sehen. Diese Vererbung sorgt dafür, dass jede Klasse verschiedene Methoden zur Verfügung hat. Dazu zählt eine Start-Methode, die beim Laden eines Objekts mit dem Skript ausgeführt wird und eine Update-Methode, die bei jeder Frameaktualisierung ausgeführt wird. Des Weiteren können weitere Methoden genutzt werden. Außerdem sorgt die `MonoBehavior` Vererbung dafür, dass diese Skripte mit Objekten in Unity verknüpft werden können.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class test : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12    // Update is called once per frame
13    void Update () {
14
15    }
16 }
```

Listing 2: Aufbau eines Unity Skriptes

## 4.4 SQL

SQL ist im Allgemeinen als Akronym für „Structured Query Language“ gesehen, obwohl SQL eine eigenständige Bezeichnung ist.<sup>24</sup> SQL wird dazu verwendet, um Daten in relationalen Datenbanken zu bearbeiten, abzufragen und anzulegen.<sup>25</sup> Die Syntax von SQL orientiert sich an der englischen Sprache. Um unabhängig von einem Datenbanksystem zu sein, wird häufig SQL verwendet, da SQL von vielen Datenbanken unterstützt wird.<sup>26</sup>

<sup>24</sup> vgl. Beaulieu, Alan[sql1] (2009) S.8

<sup>25</sup> vgl. Beaulieu, Alan[sql1] (2009) S.IX

<sup>26</sup> vgl. Adams, Ralf[sql2] (2012) S. 63 ff.

## 5 Umsetzung

Nachdem im vorherigen Kapitel die technischen Grundlagen erläutert wurden, wird jetzt auf die Umsetzung eingegangen. Dabei wird diese in drei Teile unterteilt

### 5.1 App

Google Anmeldung wird nicht genutzt, da wir nur ein paar spezielle Informationen von den Spielern brauchen und Google Daten geben?? Vorteil an Android und Google Play Store: Google Play Store: große Anzahl an vielfältigen Apps, Schutz durch Google Play, da Apps Kriterien erfüllen müssen um aufgenommen zu werden

Durchführen von Usability-Tests

### 5.2 Datenbank auf dem Handy

Wieso haben wir das gemacht? Vorteile? Nachteile? Speicherung von Daten -> wieso nicht PlayerPrefs von Unity benutzen sondern Serialisieren? Doku, Paper, ... begründen und Beispiel zeigen. Gespeichert wird in einer .dat anstatt .txt, nicht einfach lesbar bearbeitbar (daten lesen etc.)

### 5.3 Datenbank auf dem Server

Bei der Datenbank auf dem Server handelt es sich um eine MySQL Datenbank. In dieser sind die Daten der Registrierten User und deren Fortschritt in den Tabellen accounts und spielfortschritt gespeichert.



## **5.4 C# Skripte**

### **5.4.1 Kamera**

### **5.4.2 Player**

### **5.4.3 Portale**

### **5.4.4 Datenimport aus JSON**

### **5.4.5 Datenimport aus / in Datenbank**

Zum Senden der Daten der Registrierung wird das Skript SendDataToServer.cs genutzt. In diesem werden die Daten der Registrierung zwischengespeichert und am Ende an den Server gesendet. Dazu wird die Methode SendRegister() genutzt. In dieser wird die Methode RegisterUser() als Coroutine gestartet. Dazu werden zehn Parameter übergeben, der Username, die Email, das Passwort, der Vorname, der Nachname, das Geburtsdatum, das Geschlecht, der Herkunftsstaat, die native Sprache und der gewählte Charakter. Bei einer Coroutine handelt es sich um einen Thread, welcher beliebig gestartet, pausiert und beendet werden kann.

Innerhalb dieser Methode wird zu Beginn ein Hash erstellt, welcher am Server genutzt wird, um zu überprüfen, ob die Anfrage gültig ist. Dieser besteht dabei aus teilen der Eingabe und einem zusätzlichen geheimen Schlüssel, welchem nur der App und dem Server bekannt sind. Dadurch wird die Sicherheit gesteigert und es wird Angreifern erschwert unberechtigte Zugriffe auf die Datenbank zu tätigen. Bei dem Hash handelt es sich dabei um die MD5 verschlüsselten Eingabedaten. Dadurch ist es fast nicht möglich, einen Validen Hash zu bilden, ohne diese Daten zu kennen.

Nachdem der Hash erstellt wurde, werden alle Parameter in Form einer Url aneinander gehängt. Anschließend wird die URL an ein WWW Objekt übergeben und solange gewartet, bis es eine Antwort gibt. Sofern es keinen Fehler gab, wird ein Text ausgegeben, welche vom Server gesendet wird, andernfalls eine Fehlermeldung.

```

1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5  using UnityEngine.UI;
6
7  public class SendDataToServer : MonoBehaviour {
8
9      private static string secretKey = "norpg";
10     public static string registerURL = "http://norpg.it.dh-karlsruhe.de/register.php?";
11     public static string loginURL = "http://norpg.it.dh-karlsruhe.de/login.php?";
12
13     ...
14     ...
15
16     private void SendRegister() {
17         StartCoroutine(RegisterUser(userText, emailText, MD5Test.Md5Sum(passwordText),
18             firstnameText, lastnameText, birthdayText, genderText,
19             countryText, native_languageText, selected_characterText));
20     }
21
22     IEnumerator RegisterUser(string user, string email, string password, string firstname,
23         string lastname, string birthday, string gender, string country,
24         string native_language, string selected_character) {
25
26         string hash = MD5Test.Md5Sum(user + email + password
27             + firstname + country
28             + selected_character + secretKey);
29
30         string post_url = registerURL
31             + "user=" + WWW.EscapeURL(user)
32             + "&email=" + WWW.EscapeURL(email)
33             + "&password=" + WWW.EscapeURL(password)
34             + "&firstname=" + WWW.EscapeURL(firstname)
35             + "&lastname=" + WWW.EscapeURL(lastname)
36             + "&birthday=" + WWW.EscapeURL(birthday)
37             + "&gender=" + WWW.EscapeURL(gender)
38             + "&country=" + WWW.EscapeURL(country)
39             + "&native_language=" + WWW.EscapeURL(native_language)
40             + "&selected_character=" + WWW.EscapeURL(selected_character)
41             + "&hash=" + hash;
42         WWW hs_post = new WWW(post_url);
43         yield return hs_post;
44
45         if (hs_post.error != null) {
46             print("There was an error posting the high score: " + hs_post.error);
47         } else {
48             status.text = hs_post.text;
49         }
50     }
51 }

```

Listing 3: Skript SendDataToServer.cs

Auf dem Server läuft für die Datenannahme das PHP Skript register.php. Dieses nimmt die Daten aus der URL entgegen und speichert diese zunächst in Variablen ab. Anschließend wird auch in diesem Skript ein Hash gebildet und anschließend mit dem

Mitgesendetem abgeglichen. Sollte es hier einen Fehler geben, sendet der Server einen Error zurück, wenn die Hashes identisch sind, wird eine Verbindung zu der Datenbank aufgetan und ein Eintrag in der accounts Tabelle erstellt. Anschließend wird eine erfolgreiche Meldung an den Client gesendet.

Für den login innerhalb der App wird identisch vorgegangen. Dabei wird jedoch ein Datensatz in die Datenbank geschrieben, sondern nur gelsen. Des weiteren wird der Hash aus nicht so vielen Werten gebildet, da nur der Username und das verschlüsselte Passwort übermittelt werden. Nachstehend ist die Methode für den Login aus der App und der Code vom Server zur validierung zu sehen.

---

Listing 4: Skript SendDataToServer.cs

### **5.4.6 Allgemein**

## **6 Fazit und Ausblick**

### **6.1 Fazit**

### **6.2 Ausblick**

Webfrontend ...

Eigene Spiele zur Überprüfung, ob der Spieler den Standard erfüllt hat...

weitere Klassen und weiter Fächer, dementsprechend auch neue Spielwelten

Ranking/Multiplayer/...

# Anhang