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 27. April 2017

Bearbeitungszeitraum	26 Wochen
Matrikelnummer - Mehmet Ali Incekara	5672336
Matrikelnummer - Tom Wolske	1156973
Kurs	TINF14B2
Gutachter der Studienakademie	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 27. April 2017	
Ort, Datum	Tom Wolske
Karlsruhe, den 27. April 2017	
Ort, Datum	Mehmet Ali Incekara

Abstract

Zusammenfassung

Inhaltsverzeichnis

Abkürzungsverzeichnis

Abbildungsverzeichnis

Listings

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.[1] 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 realistisch aussehender Spiele zu liefern. Um den Überblick über die Vielzahl an Videospielen zu behalten, haben sich in den letzten Jahren verschiedene Plattformen etabliert, die versuchen dem Nutzer das zu bieten, was sie suchen. Dabei stellen diese viele verschiedene Arten von Spielen für die Gamer bereit, 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, neues 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 eine Schule besuchen kann. Für diesen Zweck wurde die Plattform Hone[2] entwickelt, mit der Kinder, die nicht zur Schule gehen können, die Möglichkeit haben, Wissen zu erlangen.

---

1 vgl. gesellschaftsspiele.de [**spiele**] (2015)

2 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 Web-Plattform gibt es für Kinder die Möglichkeit neue Lernspiele für verschiedene Plattformen herunterzuladen. Zusätzlich gibt es eine Ansicht der gelernten Kompetenzen.

Dieses Konzept hat zwei wesentliche Nachteile für die Benutzer. Für die Verwendung der Web-Plattform wird ein Computer benötigt und gerade weil Spiele für verschiedene Plattformen angeboten werden können, benötigt das Kind mehrere Geräte. Neben diesen Nachteilen, ist das Aussehen und die Bedienung dieser Plattform nicht reizvoll für Kinder gestaltet.

Mehr Kinder als je zuvor in der Geschichte arbeiten täglich mit immer neueren und besseren technischen mobilen Geräten. Deshalb soll eine mobile Applikation, kurz App, für Smartphones entwickelt werden, in der die Kinder auf spielerischer weise Fortschritte machen. Durch die Umsetzung als App wird den Kindern eine Plattform angeboten, mit welcher sie unabhängig und jederzeit auf die Lerninhalte zugreifen können. Ein weiterer Vorteil ist, dass die Kinder den technischen Umgang mit Smartphones lernen. Die App wird unabhängig von Hone funktionieren und es werden keine Inhalte und Funktionen übernommen.

1.2 Ziel der Arbeit

Das Ziel dieses Projektes ist es den Kindern eine Möglichkeit zu geben, mit der sie jederzeit, überall und einfach auf die Lerninhalte zugreifen und spielerisch neues Wissen erwerben können.

Dabei ist es nicht das Ziel die Lerninhalte direkt in der App abzufragen sondern Lernspiele für Smartphones anzubieten, welche es in korrekter Reihenfolge freizuschalten und zu spielen gilt.

Das Ziel dieser Arbeit ist die Vorgehensweise, Bedingungen, Probleme zu dokumentieren. Mit dieser Dokumentation soll gewährleistet werden, dass dieses Projekt von allen Verstanden wird.

1.3 Aufbau der Arbeit

Die einzelnen Kapitel dieser Arbeit repräsentieren die notwendigen Schritte, das Ziel dieses Projektes zu erreichen.

Bevor überhaupt Anforderungen spezifiziert werden können, müssen zunächst die Ideen und Gründe hinter diesem Projekt beschrieben werden. Das nächste Kapitel behandelt deshalb das Konzept der App. Zudem wird das beschriebene Konzept von anderen Spielkonzepten abgegrenzt, um die Unterschiede klarzustellen.

Im darauf folgendem dritten Kapitel wird das zum Projekt dazugehörende Software Requirements Specifikation behandelt. Dies dient zur Spezifikation der App. Neben funktionalen Anforderungen werden hier auch nicht-funktionale Anforderungen festgeschrieben.

Anschließend wird auf die technischen Grundlagen für die Umsetzung eingegangen. Dabei wird die gewählte Entwicklungsumgebung und weitere notwendigen Technologien beschrieben.

Darauf aufbauend wird im fünften Kapitel die Umsetzung behandelt. Dabei wird auf Besonderheiten und Probleme in der Entwicklungsphase eingegangen. Dafür werden die einzelnen Komponenten der App beschrieben.

Abgeschlossen wird diese Arbeit mit einem Fazit und Ausblick, in dem der weitere Werdegang des Projektes geschildert wird.

2 NoRPG

Bei NoRPG handelt es sich um eine Gamifizierung einer Lernspielplattform. Dabei soll NoRPG an ein Rollenspiel, bzw. Role Player Game (RPG), erinnern und implementiert dessen charakteristische Eigenschaften. Eine dieser Eigenschaften von RPGs ist, dass der Spielende in die Rolle realer Menschen, fiktiver Figuren, Tiere oder auch Gegenstände einer fiktiven Welt schlüpft[3]. Dabei wird eine Story erzählt, die das Kind erleben kann. Damit der Spieler allerdings in der Geschichte vorankommt, muss er verschiedene Missionen bzw. Quests erledigen. Dabei kann es sich um die verschiedensten Aufgaben handeln. 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[4].

Des Weiteren gibt es noch 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 ihnen gemeinsam spielen. Das wohl berühmteste MMORPG ist dabei World of Warcraft [5] vom Entwickler Blizzard. In MMORPGs gibt es die Möglichkeit verschiedene Events mit mehreren Spieler gemeinsam Quests zu erfüllen. Dieser Mehrspieler-Modus grenzt die MMORPGs von den RPGs ab.

Jedoch handelt es sich bei NoRPG letztendlich nicht um ein klassisches RPG oder MMORPG, sondern um eine Lernspielplattform und bietet Lernspiele zum Herunterladen an. NoRPG soll durch die Eigenschaften eines Rollenspiels die Spieler dazu anregen, weitere Lernspiele herunterzuladen und zu spielen. Deshalb wurde sich für den Namen NoRPG entschieden, da nicht alle charakteristischen Eigenschaften implementiert werden.

NoRPG ist allerdings auch nicht mit einem Massive Open Online Course (MOOC) zu verwechseln, sondern baut nur auf einem auf. Ein MOOC bezeichnet einen kostenlosen Onlinekurs. Ein MOOC würde selbst die Lerninhalte anbieten[6], wohingegen in NoRPG nur Spiele angeboten werden, die den Lerninhalt bereitstellen.

---

3 vgl. Warwitz und Rudolf [**rpgSinn**] Seite 78ff.
4 für weitere Informationen siehe `http://thewitcher.com/en/witcher3`
5 für weitere Informationen siehe `https://worldofwarcraft.com/`
6 Vgl. Porter [**moocBook**] Seite 3f.

2.1 The Global Goals

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[7]. 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.

NoRPG unterstützt dabei das Ziel, hochwertige Bildung für jeden zugänglich zu machen. Dieses Ziel wird beschrieben als Sicherung eines integrierenden Bildungssystems für alle und die Förderung von gleichberechtigten und hochwertigen lebenslangen Lernchancen[8]. 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.
- Aufbau und Weiterentwicklung von Bildungseinrichtungen, die kinder- und behindertengerecht und geschlechtsspezifisch sind und für alle eine sichere, gewaltfreie, integrative und effektive Lernumgebung bieten.

Das erste Unterziel wird in NoRPG dahingegen unterstützt, dass die Common Core State Standards implementiert werden. Da NoRPG für alle kostenfrei zugänglich ist und Mädchen und Jungen gleichberechtigt sind, werden auch diese zwei Aspekte des Unterziels unterstützt.

Da NoRPG keine Bildungseinrichtung ist wird das zweite 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 kann diese Anwendung in Bildungseinrichtungen angewendet werden. Geschlechtsspezifisch ist das Spiel nur dahingehend, dass die Kinder zu Beginn das Geschlecht ihres Charakters auswählen.

[7] vgl. Global Goals [**global**] (2017)

[8] `http://www.globalgoals.org/de/global-goals/quality-education/`

2.2 Common Core State Standards

Die Common Core sind ein Set von hochqualitativen akademischen Standards für Mathe und Englisch. Diese Lernziele skizzieren, was ein Schüler wissen und fähig sein am Ende jeder Klasse sollte. Die Standards wurden geschaffen um sicherzustellen, dass alle Schüler mit den gelernten Fähigkeiten und Kenntnissen im Collage, Karriere und im Leben, egal wo sie leben, erfolgreich zu sein.

Das Problem des Schulsystems ist, dass die Standards von Staat zu Staat variieren und stimmen meistens nicht mit dem überein, was die Kinder wissen sollten. In Erkennung der Notwendigkeit konsequenter Lernziele wurden die Common Core State Standards entwickelt[9].

Die Standards werden dabei von den zur Verfügung gestellten Spielen erfüllt. NoRPG sorgt dafür, dass diese Standards in der korrekten Reihenfolge und vollständig erfüllt werden. Betrachtet werden zunächst nur die Standards für die ersten fünf Klassen. Die Abarbeitung der Standards entspricht nicht dem klassischen Vorgehen von Schulen, so ist es beispielsweise nicht notwendig alle Standards der ersten Klasse zu erfüllen, um Stoff der zweiten Klasse freizuschalten. So ist es möglich, wenn ein Kind den Standard Geometrie der ersten Klasse vollständig erfüllt, schon weiter mit Geometrie für die zweite Klasse fortzufahren, ohne das alle anderen Mathestandards der ersten Klasse vollständig abgeschlossen wurden.

Mathe

Die Common Core State Standards konzentrieren sich auf eine klare Reihe von mathematischen Fähigkeiten und Konzepten. Die Schüler lernen Konzepte in einer organisierten Weise. Die Standards ermutigen die Schüler, reale Probleme zu lösen[10].

Das Fach Mathe lässt sich für die ersten fünf Klassen in insgesamt fünf Themenbereiche eingliedern. Zu den Themen gehören beispielsweise algebraisches Denken, Operationen im Zahlenraum bis 100, Maßeinheiten und Geometrie. Jedes dieser einzelnen Themen sind nochmals in Standards unterteilt. Erst durch diese genaue Gliederung wird es ermöglicht, dass alle wichtigen Inhalte abgedeckt werden. Damit das vorhin beschriebe Konzept umgesetzt werden kann, mussten sehr früh alle Abhängigkeiten zwischen den Standards ermittelt und visualisiert werden. Der vollständige Abhängigkeitsbaum ist im Anhang dieser Arbeit zu finden.

[9] Vgl. `http://www.corestandards.org/about-the-standards/`
[10] Vgl. `http://www.corestandards.org/Math/`

Englisch

Die Common Core State Standards bittet die Schüler, Geschichten und Literatur zu lesen, sowie komplexere Texte, die Fakten und Hintergrundwissen in Bereichen wie Wissenschaft und Sozialwissenschaften liefern. Die Schüler werden herausgefordert und gefragt, welche sie dazu bringen, auf das zurückzugreifen, was sie gelesen haben. Dies unterstreicht kritisches denken, Problemlösung und analytische Fähigkeiten, die für den Erfolg in College, Karriere und Leben erforderlich sind[11].

Das Fach Englisch lässt sich für die ersten fünf Klassen in insgesamt sechs Themenbereiche eingliedern. Zu den Themen gehört Lesen von unterschiedlichen Texten, Schreiben, Sprechen, Zuhören sowie Grammatik. Die Gliederung diese Themen entspricht den Standards. Auch hier ist der Abhängigkeitsbaum im Anhang dieser Arbeit zu finden.

2.3 Gamifizierung

Dieses Ziel der Global Goals soll dabei durch die Gamifizierung der Common Core State Standards umgesetzt werden. Gamifizierung bzw. Gamification bezieht sich auf die Analyse von spielespezifischen Eigenschaften, welche die Spiele unterhaltsam machen und diese dann in Situationen außerhalb von Spielen anzuwenden, um das Gefühl von Spaß für neue Anwendungen, wie Lernen oder Marketing, zu übertragen[12]. Ein Beispiel dafür ist PayBack. Dabei sammeln die Nutzer bei jedem Einkauf Punkte. Diese können die Kunden dann gegen Prämien eintauschen. In Videospielen sammeln die Spieler zum Beispiel Münzen um diese anschließend gegen Gegenstände einzutauschen.

Gamifizierung verwendet darüber hinaus noch weitere erfolgreiche Prinzipien aus Videospielen, um die Nutzer zu Motivieren das Produkt zu nutzen. Eines der wichtigsten Prinzipien ist die Einbettung der Handlungen in eine Geschichte. Die Erzählung einer bindenden und spannende Geschichte hilft dem Spieler sich in die Rolle seines Charakters zu versetzen und eine emotionale Bindung herzustellen. Dadurch wird das Spielerlebnis intensiver und macht dem Spieler auch mehr Spaß. Die Wichtigkeit von Spaß ist dabei nicht zu unterschätzen. Spaß motiviert die Kinder, auf eigene Faust zu lernen und das Spiel mehr zu spielen[13].

[11] Vgl. `http://www.corestandards.org/ELA-Literacy/`
[12] Umformuliert vom Oxford Dictionary
[13] Vgl. Michael und Chen [**seriousGamesFun**] Seite 40

Die Möglichkeit zu wählen ist ein weiteres wichtiges Element von Spielen. Gäbe es nicht die Möglichkeit Entscheidungen zu treffen würde es tatsächlich sich um Filme oder Bücher handeln. Diese Medien haben keine Wahlmöglichkeiten, entweder der Betrachter ist aufmerksam oder nicht. Die Wahl, in Bezug auf die Handhabung einer bestimmten Herausforderung als aktiver Teilnehmer, Kontrolle über das eigene Lernen zu haben, macht dem Spieler die eigenen Entscheidungen und deren Konsequenzen bewusster[14].

Ein weiteres Prinzip welches oft verwendet wird ist direktes Feedback. Dieses trifft in den meisten Fällen zusammen mit dem Prinzip von Belohnungen auf. Es ist schwer, sich das Leben ohne jegliche Rückmeldung vorzustellen, in der Tat ist es unmöglich. Rückmeldungen sind Informationen welche den Menschen helfen die Welt um uns herum besser zu verstehen und dies ist entscheidend für die Weiterentwicklung. Durch zusätzliche Belohnungen für besondere bzw. korrekte Leistungen wird dieser Effekt verstärkt[15]. Allerdings ist zu beachten, dass die Belohnungen einen Nutzen für den Spieler darstellen, damit es sich lohnt Aufwand für diese Belohnungen aufzuwenden[16]. Vervollständigt werden diese Prinzipien durch die Implementierung Spielstatus durch Level und Auszeichnungen.

Zwei weitere Prinzipien, die Onlinespiele ausmachen, ist zunächst der ständige Wettbewerb mit anderen Spieler und Möglichkeit Aktivitäten zusammen in einem Team mit anderen Spieler zu bewältigen. Darüber hinaus sind die Faktoren wie Erfolgserlebnisse, Gruppenzugehörigkeit und soziale Akzeptanz wichtig.

Im nächsten Unterkapitel wird die Idee zur Umsetzung des Prinzipes der Einbettung der Handlungen des Spielers in eine Geschichte behandelt.

2.4 Die Geschichte

In NoRPG spielt der Spieler einen Charakter, der in dem Dorf Rutherglen wohnt. Am Anfang der Geschichte wird dieses Dorf von dem Bösewichten heimgesucht. Dieser klaut alle Farben das komplette Dorfes und alles wirkt farblos und somit auch emotionslos. Nun hat der Spieler, das Ziel die Farben wieder zurück zu bringen. Dazu muss der Spieler in verschiedenen Welten gehen und verschiedene Quests erfüllen.

Die einzelnen Welten repräsentieren eine Klassenstufe und sind dementsprechend anspruchsvoll und unterschiedlich gestaltet. Die Standards der ersten Klasse sind in der

[14] Vgl. Routledge **[seriousGamesPrinciples]** Seite 30f.
[15] Vgl. Routledge **[seriousGamesPrinciples]** Seite 32ff.
[16] Vgl. Berkling, Faller und Piertzik **[gamesPaper]** Seite 8

ersten Welt. Äquivalent verhält es sich auch in den anderen Welten. Neben den Standards gibt es noch Truhen zu finden, die den Spieler in der Geschichte von NoRPG voranbringen. Jede Welt ist dabei in Unterbereiche gegliedert, die der Spieler mit der Zeit erreichen kann. Dabei wird der zu erkundende Bereich immer größer, je weiter der Spieler in der Geschichte vorankommt.

Eine sehr markante Eigenschaft von RPGs ist, dass die Spielwelt offen gestaltet ist und dem Spieler nur eine ganz grobe Vorgehensweise vorgeben wird. Der Spieler kann sich frei in der jeder Welt bewegen, kann sich mit allen Bewohnern unterhalten und verschiedene Sachen, wie beispielsweise Ruinen, etc., entdecken. Dies wird als Open World bezeichnet. Dem Spieler werden nur Grenzen vorgegeben in denen das Kind seine eigene Herangehensweise entwickeln kann. Eine grobe Vorgabe ist notwendig, damit das lernen strukturiert stattfindet.

Startwelt: Das Dorf Rutherglen

Rutherglen ist die Heimat des Spielers und dient als Brücke zwischen allen Welten. In die verschiedenen Welten gelangt der Spieler über Portale, die in Rutherglen verteilt sind. Bei dem Heimartort des Spielers handelt es sich um ein verschlagenes, unscheinbares und ruhiges Dorf. Der Spieler kann in dieser Welt mit allen Bewohnern interagieren und sprechen. Diese erzählen Geschichten über das Dorf, über ihr Leben oder geben Tipps und Hinweise.

Welt 1: Die dichten Wälder Talhan

Talhan ist das Waldgebiet von Rutherglen. In dieser Welt ist der Wald das primäre Element. Dieser ist in verschiedenen Ausprägungen vorhanden. Das Kind findet sich in einem großen Wald mit verschiedenen Baumarten wieder: Von Laubbäumen, über tropische Bäume bis hin zu großen Fichtenbäume. Neben diesen Elementen gibt es noch weitere Sachen zu entdecken, die die Reise des Kindes durch die Wälder Talhan spannender gestalten soll. Bei Abschluss dieser Welt kehrt auch die erste Farben wieder zurück in das Dorf. Dabei handelt es sich um die Farbe Grün.

Welt 2: Die tropischen Inseln von Galapagos

Bei der nächsten Welt handelt es sich um die tropischen Inseln von Galapagos. Diese Welt ist im Gegensatz zum Wald sehr farbenfroh, allerdings gibt es auch viel Wasser. Galapagos besteht dabei aus mehreren Inseln, welche mit dem Schiff erreicht werden können. Auch hier gibt es auf den verschiedenen Inseln unterschiedliche Sachen zu

entdecken. Bei Abschluss dieser Welt kehrt nun auch die nächste Farbe wieder zurück in das Dorf. Dabei handelt es sich um die Farbe Blau.

Welt 3: Die endlose Wüste Kalahari

Kalahari ist eine sehr große Wüste und ist trostloser wie die vorherigen Welten. In dieser Welt wird der Begriff Open World sehr deutlich. Im Vergleich zu fast allen anderen Welten gibt es keinen Weg oder Wegweiser. Dieses Open World Konzept hat den Sinn, das Gefühl einer endlosen riesigen Wüste zu transportieren. Innerhalb dieser Wüste kann das Kind Oasen, verschiedene Ruinen und ägyptische Wahrzeichen, wie beispielsweise Pyramiden, entdecken. Auch hier erhält der Spieler bei Abschluss die nächste Farbe. Dabei handelt es sich um die Farbe Gelb.

Welt 4: Das verschneite Gebirge Lhotse

Die vorletzte und größte Welt sind die verschneite Gebirge Lhotse. In dieser Eiswelt dreht sich alles um Eis und Schnee. Neben der Wüste ist diese Welt die zweite Welt, in der das Prinzip Open World sehr deutlich wird. Es gibt viele hohe Berge, verschiedene große Höhlen und weitere Sachen. Nachdem das Kind alle Truhen auch in dieser Welt gefunden hat erhält es die vorletzte Farbe Türkis.

Welt 5: Der Vulkan Ätna

Die zunächst letzte Welt ist Ätna. Hier dreht sich alles um Feuer. In dieser Welt gibt es verschiedene Inselplattformen, welche durch Lava getrennt sind. Allerdings sind einige untereinander mit Brücken verbunden. Auf den verschiedenen Inseln kann der Spieler verschiedene Dinge erkunden, darunter Drachen, Vulkane oder Ruinen. Nach dem das Kind auch diese Welt bewältigt, sind alle Farben wieder zurück im Dorf. Aus dieser Welt bekommt das Kind die fehlende letzte Farbe Rot.

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 beschriebenen wird. Einige Kapitel des SRS werden allerdings nicht behandelt, da sie keine Relevanz für NoRPG haben oder an einer anderer 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[17].

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[18]. Zudem dient die Spezifikation zur Kommunikation zwischen den Stakeholdern und den Entwicklern.

3.1.2 Umfang

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

[17] vgl. Tripp [**srsIEEE**](1998) Seite 3
[18] vgl. Rozanski [**rozanski2011**](2011) Seite 6

3.2 Allgemeine Beschreibung

Im zweiten Kapitel des SRS werden alle 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 unterschiedlichsten weisen 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.

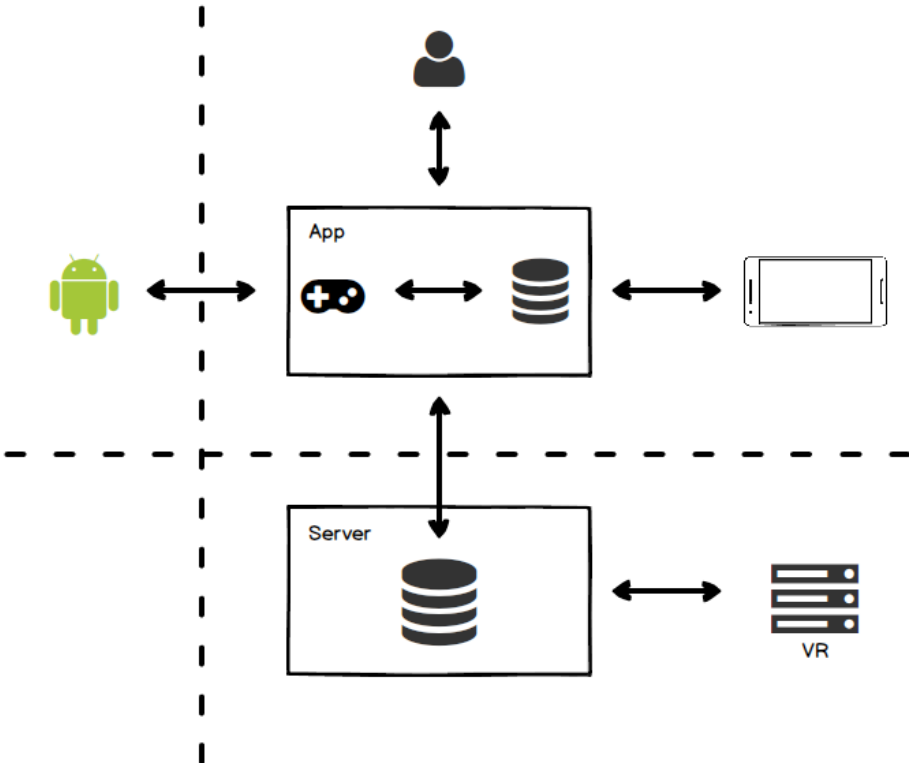

Abbildung 1: High-Level-View von NoRPG

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 virtualisierten 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[19].

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.

[19] 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-Systems. Dazu zählt die Bedienung der Android-Oberfläche und die des Google Play Stores. Zudem sollten die User englische Texte lesen und verstehen können, da NoRPG zunächst nur in Englisch 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[20]. 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

[20] vgl. Dan Morrill (Google) [**google1**]

angeordnet sind. Diese Eigenschaft gilt ebenfalls für alle anderen Benutzerschnittstellen und wird nicht bei den Beschreibungen der anderen User Interfaces zusätzlich 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.

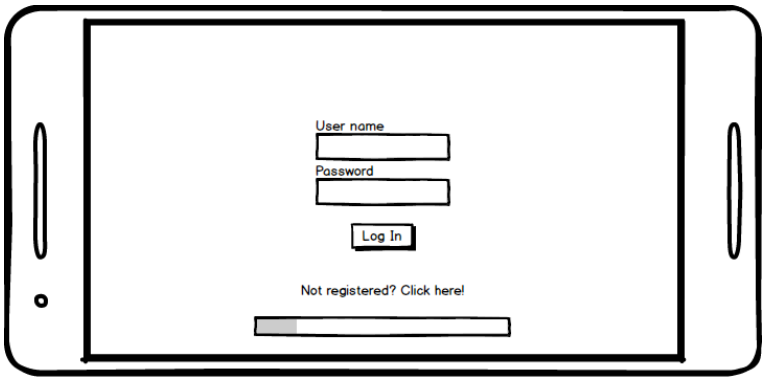

Abbildung 2: Mockup: Login-Screen

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 Aussehen seines Charakters.

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,

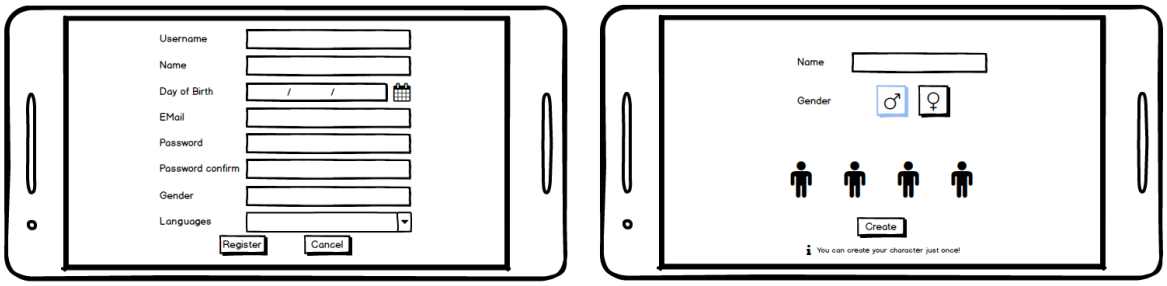

Abbildung 3: Mockups: Registrierungsformular und Charaktererstellung

die Spieler über viele Spiele erkennen. Die meisten von ihnen sind statisch auf dem Bildschirm, so dass sie während des Spiels sichtbar bleiben.

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 Bildschirmes 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

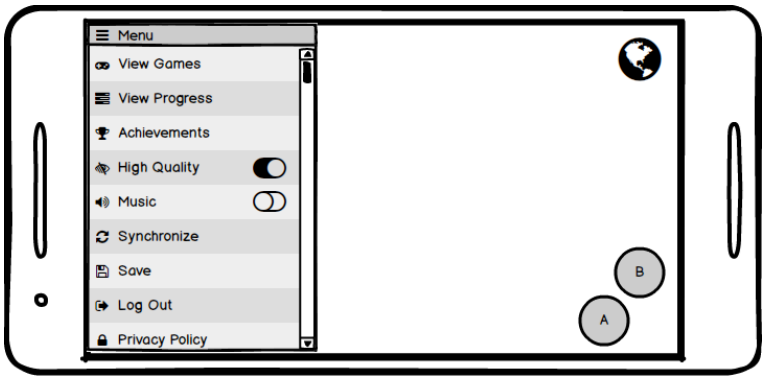

Abbildung 5: Mockup: Menü

Abbildung 6 als Mockups zu betrachten.

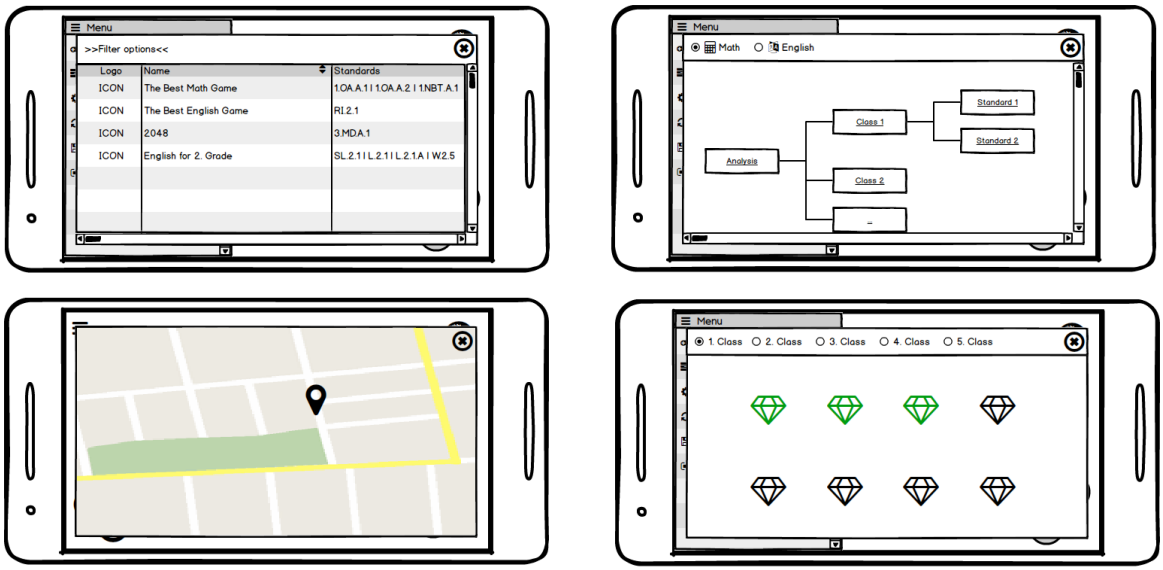

Abbildung 6: Mockups: Spielliste, Fortschrittanzeige, Karte und Erfolgsübersicht

Hardwareschnittstellen

Schnittstellen zwischen NoRPG und Hardwarekomponenten werden als Hardwareschnittstellen 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. Se 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 Hardwareschnittstelle.

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.

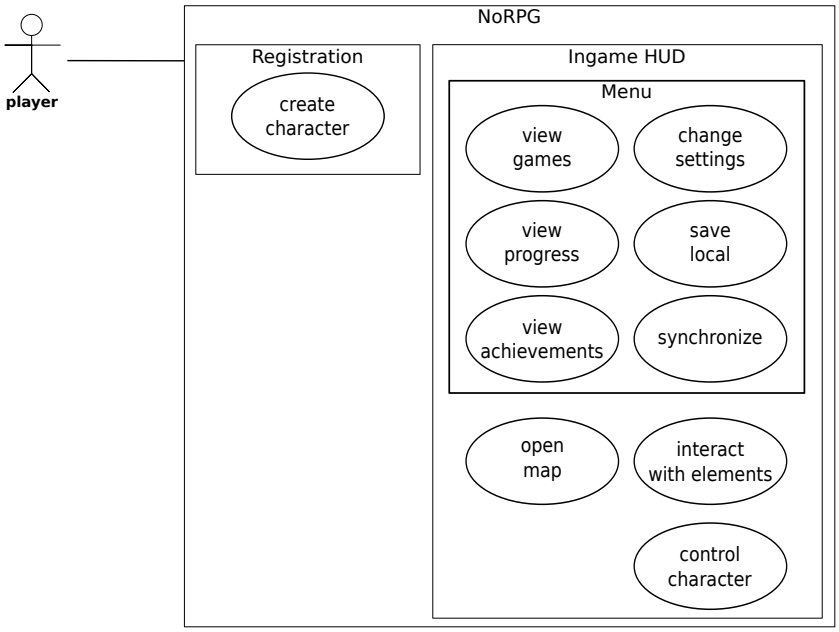

Abbildung 7: Overall Use Case Diagramm

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 beschrieben 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.

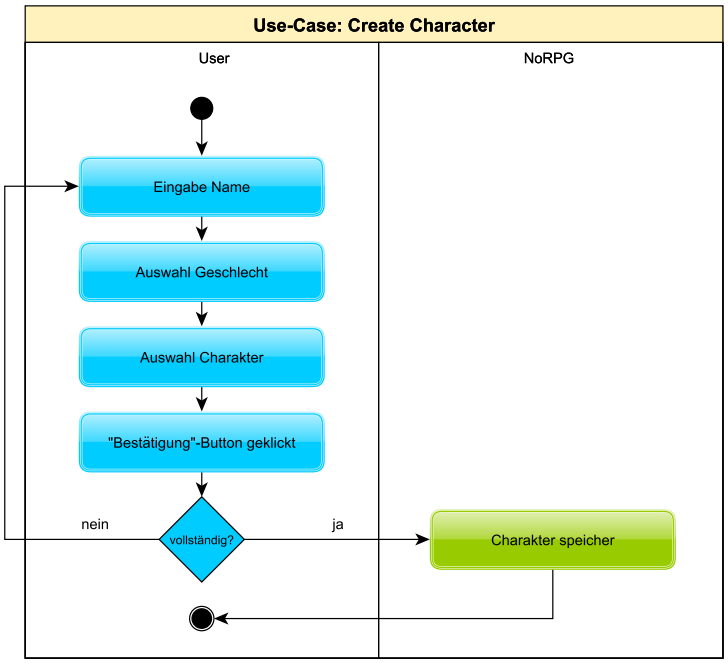

Abbildung 8: Activity Diagramm: Create Character

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 existiert. 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 Acitivity Diagramm in Abbildung 9) zeigt den vollständigen Ablauf.

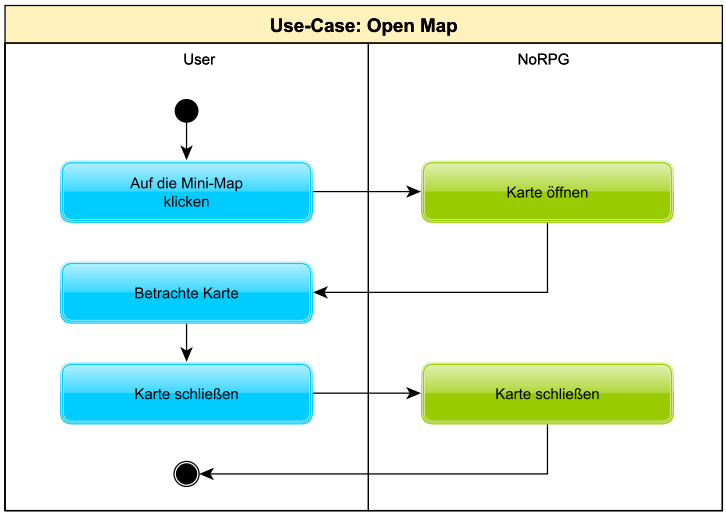

Abbildung 9: Activity Diagramm: Open Map

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.

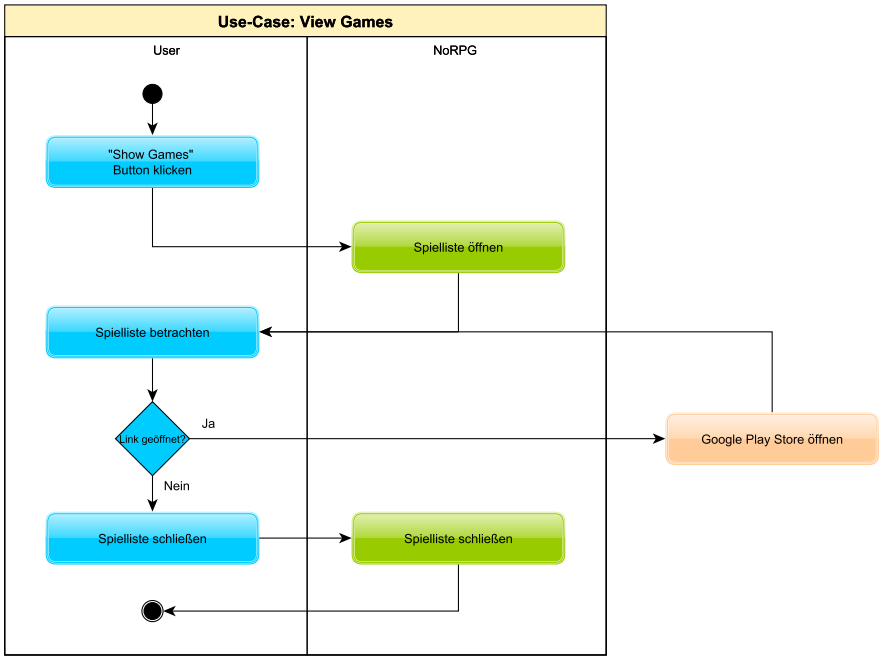

Abbildung 10: Activity Diagramm: Show Games

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.

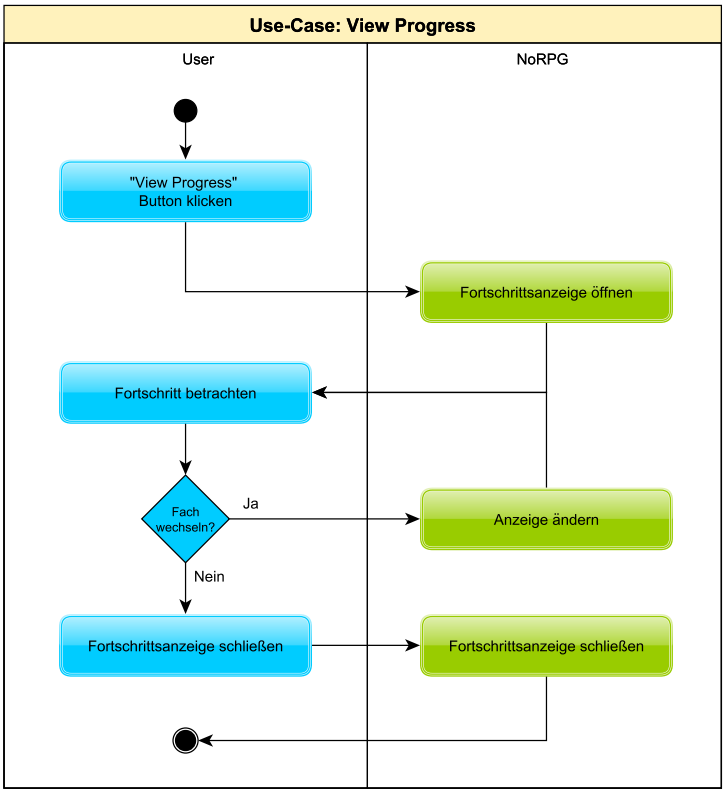

Abbildung 11: Activity Diagramm: View Progress

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 Anwendungsfalls, 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 Fort-

schritt in NoRPG. Hier kann der Spieler zwischen den Welten wechseln anstatt zwischen den Fächern und es wird keine Baumstruktur präsentiert.

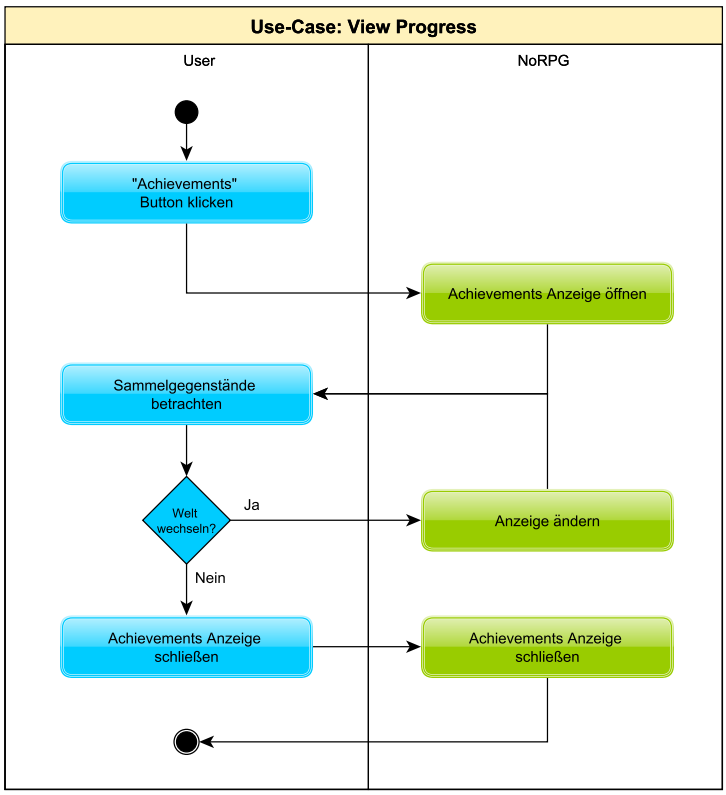

Abbildung 12: Activity Diagramm: Achievements

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 Qualityän, 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.

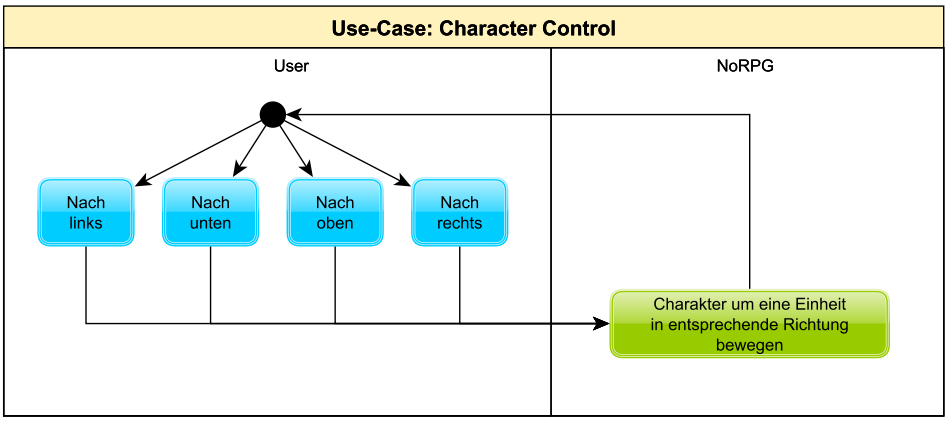

Abbildung 13: Activity Diagramm: Character Control

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 einen Element im Spiel interagieren möchte. Dafür hat der Spieler 2 Tasten auf dem HUD (vgl. Ab-

bildung 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 Ä"Button gestartet werden. Wenn ein Element zur Interaktion vorhanden ist startet erst die Interaktion. Anschließend kann der Spieler mit dem Äöder "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 Ä"Button wird die Interaktion fortgeführt bis keine Aktion mehr vorhanden ist, also in unserem Händlerbeispiel keine Unterhaltung mehr gibt.

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.

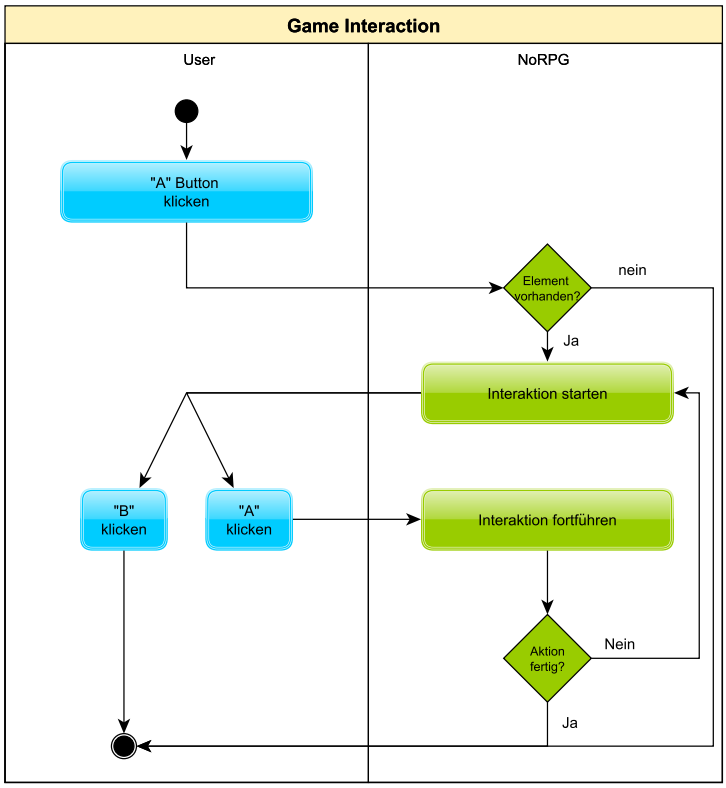

Abbildung 14: Activity Diagramm: Game Interaction

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[21].

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.

[21] vgl. Andrew Lee [**performance1**]

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 gegeben 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.

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. Zugegriffen 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[22]. Für eine Internetverbindung bietet Samsungs Galaxy S4 eine drahtlose Schnittstelle, die nach IEEE 802.11a/ac/b/g/n funktioniert.

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 In-

[22] vgl. Google **[s4Google]**

formationen aber auch die Anpassung softwaregesteuerten 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 [23].

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, das so wenig Berechtigungen wie notwendig verlangt werden[24].

Es gilt, desto einfach die App gestaltet ist, desto mehr Kinder verstehen diese. Ein Beispiel dafür ist die Charaktersteuerung. Moderne Spiele haben keine sichtbaren Steuerkreuz, 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 verwedent wird. Navigation soll schnell erkennbar und nachvollziehbar sein.

[23] vgl. Wendy B. **[intelKids]**
[24] vgl. klick-tipps.net **[appsforkids]**

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[25].

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.

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

[25] vgl. Becky White [**smashMagazin**]

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 vergleichen 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 vorhanden 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 Genration erhältlich ist[26], 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.

[26] vgl. Philippe Fischer, Michael Huch **[computerBildSamsung]**

4 Technische Grundlagen

Im folgenden Abschnitt werden die technischen Grundlagen behandelt. 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. Dazu können in Unity 3D-Projekte, aber auch 2D beziehungsweise 2,5D Projekte umgesetzt werden. Bei einer 2,5D Anwendung handelt es sich um eine Mischung von 2D und 3D Elementen. Dank Unity sind diese Projekte nach der Entwicklung plattformübergreifend einsetzbar.[27] Die Entwicklungsumgebung teilt sich dabei in verschiedene Bereiche auf.

Das „Scene" Fenster ist in den Standardeinstellungen in der Mitte des Bildschirms zu sehen(Siehe Abbildung 15)[28]. 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[29]. Diese variieren je nach gewähltem Objekt. Auch hier können die Position oder Skalierung eines Objektes verändert werden, allerdings werden hier auch darüber hinausgehende visuellen sowie die physischen Eigenschaften der Objekte verändert.

Objekte können des weiteren im „Hierarchy" Fenster[30] ausgewählt werden. In diesem Fenster werden alle Objekte der aktuellen Szene aufgelistet. Dabei wird zwischen unterschiedlichen Ebenen unterschieden, so dass genau gesehen werden kann welche Objekte zusammengehören.

---

27 vgl. Unity3D [**unity1**] (2017)
28 vgl. Unity3D [**unity2**] (2017)
29 vgl. Unity3D [**unity3**] (2017)
30 vgl. Unity3D [**unity4**] (2017)

Abbildung 15: Darstellung von Tablemappings

Um alle Dateien zu sehen, die zu einem Projekt gehören, gibt es das „Project" Fenster[31]. 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[32]. Hier kann das fertige Projekt angesehen und getestet 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 Umgenag mit Unity sind.

31 vgl. Unity3D **[unity5]** (2017)
32 vgl. Unity3D **[unity6]** (2017)

Prefabs

Bei Prefabs handelt es sich um fertige Objekte, die in Szenen verwendet werden können[33]. 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[34]. 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.

4.1.1 Grafische Oberflächen mit Unity

Die grafische Oberfläche wird mit Unity's zur Verfügung gestelltem UI System implementiert. Der Canvas, zu deutsch Leinwand, stellt dabei die Hauptkomponente bzw. Elternkomponente dar. Alle UI Elemente sind Kinder des Canvas und erben somit die Eigenschaften des Canvas. Unity bietet standardmäßig eine große Anzahl an UI Elementen, wie beispielsweise Buttons, Toggle-Buttons, Dropdown-Listen und Eingabefelder[35]. Der Canvas verwendet das EventSystem Objekt von Unity. Das Event System ermöglicht es, alle Touchscreen-Eingaben an die korrekten UI Elemente in der App weiterzuleiten[36].

Bei der Implementierung des Canvas wird zwischen drei unterschiedlichen Render-Modi unterschieden[37]. Je nachdem welcher Modus verwendet wird, ist die grafische Oberfläche anders in das Spiel integriert.

Screen Space - Overlay

Standardmäßig ist jedes Canvas als Screen Space - Overlay implementiert. Dieser Modus platziert die UI Elemente auf den Bildschirm über die komplette Szene. Die Benutzeroberfläche wird über alle anderen Grafiken, wie die Kamera-Ansicht, gezogen. Wenn die Auflösung geändert wird, passt der Canvas die Größe aller Kindelemente automatisch an die neue Auflösung an. Deshalb ist es besonders wichtig, dass alle UI Elemente unter dem Canvas hängen.

[33] vgl. Unity3D [**unity7**] (2017)
[34] vgl. Unity3D [**unity8**] (2017)
[35] Vgl. `https://docs.unity3d.com/Manual/UIInteractionComponents.html`
[36] Vgl. `https://docs.unity3d.com/Manual/EventSystem.html`
[37] Vgl. `https://docs.unity3d.com/Manual/UICanvas.html`

Die folgende Grafik[38] 16 verdeutlicht, dass alle Elemente des Spieles hinter der Benutzeroberfläche positioniert sind.

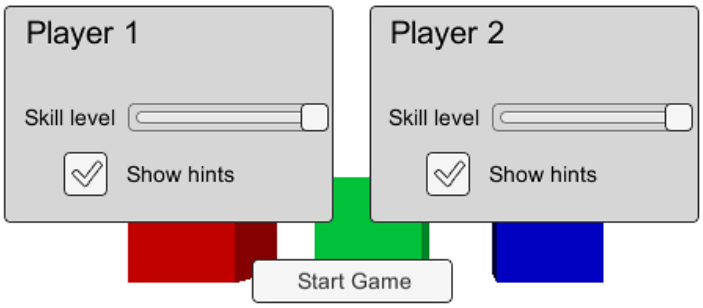

Abbildung 16: Render-Modus: Screen Space - Overlay

Die wesentliche Nachteile vom Screen Space - Overlay ist, dass der sichtbare Bereich des Spieles durch die UI Elemente verdeckt wird. Deshalb ist es besonders wichtig nur die wichtigsten Elementen immer sichtbar im Vordergrund zu lassen. Dies hat Auswirkungen auf die Positionen, Größe, Farbe und Design. Ein weiterer Nachteil ist, dass Spielelemente nicht als Teil des User Interfaces verwendet werden können.

Auf der Gegenseite können keine UI Elemente oder wichtige Informationen von den Gegenständen im Spiel verdeckt werden und ein weiterer Vorteil ist, dass die Position auf dem Bildschirm immer an die Auflösung angepasst wird und sich nicht ändert, unabhängig davon welche Aktionen der Spieler im Spiel ausführt oder auf welcher Hardware NoRPG gespielt wird.

Screen Space - Camera

Dieser Modus ähnelt dem ersten Modus sehr, aber in diesem Render-Modus wird der Canvas eine vorgegebene Distanz vor einer bestimmten Kamera platziert. Die UI Elemente werden von der bestimmen Kamera gerendert, was bedeutet, dass die Kameraeinstellungen das Erscheinungsbild der Benutzeroberfläche beeinflussen. Wenn sich die Auflösung des Bildschirms verkleinert, oder die Kameraposition und -winkel sich ändern, ändert sich der Canvas automatisch mit und passt die Größe der Elemente an.

Der größte Unterschied ist allerdings, wenn 3D-Objekte in der Szene näher an der Kamera sind als die UI Elemente, werden diese auch auf dem Bildschirm vor der Benutzeroberfläche platziert. Dieses Verhalten wird in der folgenden Grafik[39] 17 deutlich.

[38] Bild entnommen von `https://docs.unity3d.com/Manual/class-Canvas.html`
[39] Bild entnommen von `https://docs.unity3d.com/Manual/class-Canvas.html`

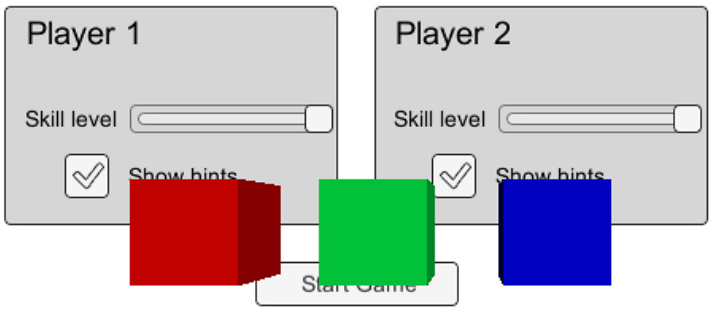

Abbildung 17: Render-Modus: Screen Space - Camera

Wenn die 3D-Objekte näher als die UI-Elemente sind können Informationen und Elemente, wie Buttons, verdeckt werden und die Interaktion mit diesen Elementen ist nicht mehr möglich. Allerdings ermöglicht dieser Modus, dass Spielelemente als Teil der UI, auch wenn nur kurzfristig, funktionieren können. Trotz diesen Vorteils ist dieses Verhalten komplexer und fehleranfälliger.

World Space

In diesem Render-Modus wird der Canvas wie jedes andere Objekt in der Szene behandelt. Die Größe und Position kann manuell gesetzt werden und die UI Elemente werden basierend auf die 3D-Position entweder vor und hinter 3D-Objekten der Szene platziert. Dies ist nützlich für UIs, die dazu bestimmt sind, ein Teil der Welt zu sein. Dies wird auch als s̈terblicheB̈enutzeroberfläche bezeichnet, da diese nicht immer im Vordergrund sein muss. Im Gegensatz zum Screen Space - Camera Modus muss die UI Ebene nicht vor die Kamera platziert werden.

Die Größe des Canvas kann manuell eingestellt werden, aber seine Größe auf dem Bildschirm wird von dem Winkel und der Entfernung der Kamera abhängen. Andere Szeneobjekte können hinter, mittendrin oder vor dem Canvas sein. Dieses Verhalten wird in folgender Grafik [40] 18 deutlich.

Vorteile dieses Modus sind, dass alle 3D Elemente als Teil der UI verwendet werden können. Nachteile ...

[40] Bild entnommen von `https://docs.unity3d.com/Manual/class-Canvas.html`

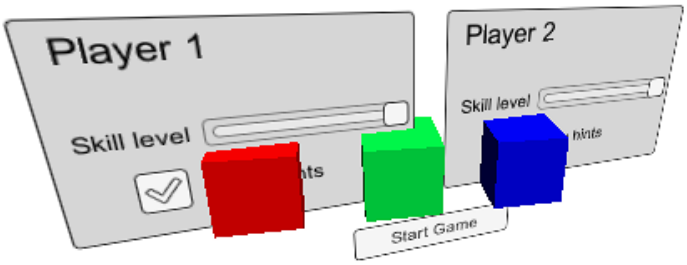

Abbildung 18: Render-Modus: World Space

4.2 Visual Studio

Mit Microsoft Visual Studio ist es möglich in verschiedenen Programmiersprachen zu programmieren[41]. 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.[42]

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.

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.

[41] vgl. Microsoft [**microsoft1**] (2015)
[42] vgl. Microsoft [**microsoft2**] (2015)

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# Skripts 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.[43] SQL wird dazu verwendet, um Daten in relationalen Datenbanken zu bearbeiten, abzufragen und anzulegen.[44] 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.[45]

[43] vgl. Beaulieu, Alan[**sql1**] (2009) S.8
[44] vgl. Beaulieu, Alan[**sql1**] (2009) S.IX
[45] 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

Nachfolgend wird auf die einzelnen C# Skripte eingegangen, welche für die App benötigt werden. Darüber hinaus wird auf weitere, für die Skripte essentielle umgesetzte Teile der App eingegangen.

5.4.1 Player

Der Player setzt sich aus mehreren einzelnen Objekten zusammen, darunter die Textur für den Spieler, ein Teil deiner Minimap und ein weiteres leeres Objekt, welches für die Kamera wichtig ist. An dem Player selbst befinden sich wiederum die Skripte und von Unity vorgefertigten Objekte. Ein sogenannter Rigedbody sorgt dafür, dass der Player Gravitation erfährt und nicht einfach durch die Luft schweben kann. Ein Character Controller fügt Eigenschaften wie eine Größe und höhe ein. Darüber hinaus besitzt der Player einen Animatoir. Dieser ist zusammen mit dem ThirdPersonController dafür verantwortlich, dass sich der Player in der Szene bewegt und die Animationen korrekt ausgeführt werden. Damit dies passiert, wird diesem Controller der Wert speed und direction übergeben. Daraus resultieren folgene Möglichkeiten der Animation

speed	direction	Ergebnis
0	0	Animation Idle
>0	0	Animation Walk
>0	0.3	Animation Walk Right Short
>0	0.5	Animation Walk Right Medium
>0	-0.3	Animation Walk Left Short
>0	-0.5	Animation Walk Left Medium
>0.5	0	Animation Run
>0.5	0.3	Animation Run Right Medium
>0.5	0.5	Animation Run Right Wide
>0.5	-0.3	Animation Run Left Medium
>0.5	-0.5	Animation Run Left Wide

Tabelle 1: Mögliche Animationen je nach Wert speed und direction

Durch diese vielzahl an Animationen ist gewährleistet, dass der Player zu jeder Zeit die richtige Animation ausführt und die Bewegung nicht unnatürlich aussieht. Die Werte speed und direction für diese Animation kommen dabei aus dem Skript CharacterControll.cs . Diese ist für die Steuerung des Players zuständig. Hier wird die Toucheingabe über den Joystick in Weltkoordinaten umgewandelt und der Player beginnt sich zu bewegen. Dazu ist die Update Methode genutzt worden. In dieser wird, sofern ein Animator an dem Player vorhanden ist, die horizontale und vertikale Bewegung des Joysticks in die direction und den speed umgewandelt. Das ganze passiert

dabei in der Methode StickToWorldspace, welche die Transform vom Player, der Kamera, die direction und den speed, übergeben bekommt. Nachdem die Methode aufgerufen wurde und abgeschlossen ist, werden die Werte von direction und speed an den ThridPersonController übergeben und die Animation startet, wie zuvor beschrieben.

In der StickToWorldspace Methode wird zu Begin die rootDirection gesetzt, welche sich dabei aus der Z-Achsen Koordinate zusammensetzt. Anschließend wird die stickDirection gesetzt, durch den horizontalen und vertikalen Wert des Joysticks. Nun wird durch das quadrieren der beiden Werte der speed berechnet. Dieser liegt dabei zwischen Null und Eins.

Danach das ganze in bezug zu der Positionsrichtung der Kamera gesetzt, um die Bewegungsrichtung zu erhalten, um anschließend das Kreutzprodukt aus diesen beiden Werten zu berechen. Mit hilfe dessen kann bestimmt werden, ob sich der Player nach Rechts oder nach Links bewegen soll.

```
using UnityEngine;
using CnControls;
using UnityStandardAssets.CrossPlatformInput;
using UnityEngine.SceneManagement;

public class CharacterControll : MonoBehaviour {

        ...

    private float speed = 0.0f;
    private float direction = 0f;
    private float horizontal = 0.0f;
    private float vertical = 0.0f;
    private AnimatorStateInfo stateInfo;

        ...

    void FixedUpdate() {
        if(IsInLocomotion() && ((direction>=0 && horizontal>=0)
                || (direction <0 && horizontal < 0))) {

            Vector3 rotationAmount = Vector3.Lerp(Vector3.zero, new Vector3(0f,
            rotationDegreePerSecound * (horizontal < 0f ? -1f : 1f), 0f),
            Mathf.Abs(horizontal));

            Quaternion deltaRotation = Quaternion.Euler(rotationAmount * Time.deltaTime);
            this.transform.rotation = (this.transform.rotation * deltaRotation);
        }
    }

    public bool IsInLocomotion() {
        return stateInfo.nameHash == m_LocomotionID;
    }

    void Update() {

        if (animator) {
            stateInfo = animator.GetCurrentAnimatorStateInfo(0);
            horizontal = CnInputManager.GetAxis("Horizontal");
            vertical = CnInputManager.GetAxis("Vertical");
            StickToWorldspace(this.transform, gamecam.transform, ref direction, ref speed);
            animator.SetFloat("speed", speed);
            animator.SetFloat("direction", direction, directionDumpTime, Time.deltaTime);
        }
    }

    ...

    public void StickToWorldspace(Transform root, Transform camera,
        ref float directionOut, ref float speedOut) {

        Vector3 rootDirection = root.forward;
        Vector3 stickDirection = new Vector3(horizontal, 0, vertical);
        speedOut = stickDirection.sqrMagnitude;
        Vector3 CameraDirection = camera.forward;
        CameraDirection.y = 0.0f;
        Quaternion referentialShift = Quaternion.FromToRotation(Vector3.forward,
                Vector3.Normalize(CameraDirection));
```

```
59
60          Vector3 moveDirection = referentialShift * stickDirection;
61          Vector3 axisSign = Vector3.Cross(moveDirection, rootDirection);
62          float angleRootToMove = Vector3.Angle(rootDirection, moveDirection) *
63                  (axisSign.y >= 0 ? -1f : 1f);
64
65          angleRootToMove /= 180f;
66          directionOut = angleRootToMove * directionSpeed;
67      }
68  }
```

Listing 3: Skript CharacterController.cs

5.4.2 Kamera

5.4.3 Portale

5.4.4 Minimap

5.4.5 Interaktionsmöglichkeiten

Chests, Händler, etc....

5.4.6 Pfadfindungssystem Schiff

Die tropischen Inseln von Galapagos werden im Spiel mit einem Schiff bereist. Dabei kann der Spieler aus fünf verschiedenen Inseln auswählen, und je nach Insel fährt das Schiff einen anderen Pfad. Diese Pfade sind dabei statisch und festgelegt. Es gibt dabei 20 Pfade, vier von jeder Insel. Jeder dieser Pfade ist ein Gameobject mit dem Skript EditorPathScript. (siehe Listing 4)

```
1
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5
6  public class EditorPathScript : MonoBehaviour {
7
8      //Color of the lines betwwen the single points
9      public Color rayColor = Color.white;
10     //List of all points of the path
11     public List<Transform> path_objs = new List<Transform>();
12     //List of all transformobjects of every point in path
13     Transform[] transforms;
14
15     //Method to draw in the Unity Editor
16     void OnDrawGizmos () {
17         //Set Color of lines to selected color
18         Gizmos.color = rayColor;
19         //Get all transformobjects of childrends
20         transforms = GetComponentsInChildren<Transform>();
21         //Clear the list
22         path_objs.Clear();
23
24         //Foreach transformobject at it to list if it is not the parentobject
25         foreach (Transform path_obj in transforms) {
26             if (path_obj != this.transform) {
27                 path_objs.Add(path_obj);
28             }
29         }
30         //Draw lines between every Point
31         for(int i = 0; i < path_objs.Count; i++) {
32             Vector3 position = path_objs[i].position;
33             if (i > 0) {
34                 Vector3 previos = path_objs[i - 1].position;
35                 Gizmos.DrawLine(previos, position);
36                 Gizmos.DrawWireSphere(position, 0.3f);
37             }
38         }
39     }
40 }
```

Listing 4: Skript EditorPathScript.cs

Dieses sorgt dafür, das die Pfade grafisch im Unity Editor angezeigt werden. Dieses Skript zeichnet dabei von einem Startpunkt zum nächsten eine Linie, solange, bis der letzte Punkt erreicht wird. Dadurch wird es möglich die Pfade grafisch zu bearbeiten und erleichtert so die Arbeit. Nun wird beschrieben, wie das Schiff dem Pfad folgt, sobald der Player den Pfad ausgewählt hat. Dabei wird auf den Pfad von Insel eins zu zwei eingegangen. Für alle anderen Pfade ist das vorgehen identisch.

Über ein Userinterface wählt der Player den betroffenen Pfad aus. Dabei wird dieser über einen Button ausgewählt. Sobald dieser geklickt wurde wird eine Funktion in Listing 5 ausgeführt. Diese selektiert den gewünschten Pfad aus den verfügbaren Pfaden

und setzt den boolean playerOnShip auf true.

```
1
2  public void selectFirstClass () {
3  //if player is not on the same Island
4   if (lastButtonClicked != "1") {
5    hud.SetActive(false);
6    // set path to the island on that the player is to island 1
7    path = GameObject.Find("Path" + lastButtonClicked + "_1").GetComponent<EditorPathScript>();
8    lastButtonClicked = "1";
9    playerOnShip = true;
10  }
11 }
```

Listing 5: Methode selectFirstClass

Da dieser boolean nun auf true gesetzt ist, wird der Code der Methode playerGetOnShip ausgeführt. Der Code ist in Listing 6 zu sehen. Durch diese Methode bewegt sich der Player mit dem Schiff in der selben Geschwindigkeit. Darüberhinaus blokiert diese Funktion die Eingabemöglichkeiten des Players während der Fahrt.

```
1
2  public void playerGetOnShip () {
3   if (!playerOnShip) {
4    cc.enabled = false;
5
6    player.GetComponent<Animator>().SetFloat("speed", 0.0f);
7    player.GetComponent<Animator>().SetFloat("direction", 0.0f);
8    player.transform.position = ship.transform.position;
9
10   player_mesh.SetActive(false);
11   player_text.SetActive(false);
12
13   follow.transform.parent = ship.transform;
14   follow.transform.position = ship.transform.position + new Vector3(2.25f, 18.07f, -7.07f);
15
16   minidot.SetActive(false);
17
18   hud.SetActive(true);
19  }
20 }
```

Listing 6: Methode playerGetOnShip

Des Weiteren wird die Funktion moveShip ausgeführt. Diese ist für die eigentliche Bewegung des Schiffes verantwortlich. Sobald playerOnShip auf true gesetzt ist, wird zuerst die Distanz zwischen dem ersten Punkt des Pfades und des Schiffes gespeichert und die Position des Schiffes an diese stelle verschoben. Anschließend wird die drehung des Schiffes an die Fahrtrichtung angepasst und gedreht. Sobald die Distanz zwischen der aktuellen Position des Schiffes und dem nächsten Punkt des Pfades kleiner ist wie eine minimale Distanz, wird die currentWayPointId um eins erhöhrt. Solange dieser Wert nicht größer ist wie die Anzahl der Punkte des Pfades werden diese Schritte wiedserholt und das Schiff bewegt sich entlang des Pfades.

```
1
2  public void moveShip () {
3   //if player is on the ship
4   if (playerOnShip) {
5    //get the distance between ship and the first point
6    float distance = Vector3.Distance(path.path_objs[currentWayPointId].position, transform.position);
7    //set the ship to the first point
8    transform.position = Vector3.MoveTowards(transform.position, path.path_objs[currentWayPointId].position, Tim
9    //set the player on ship
10   player.transform.position = transform.position;
11
12   //rotate the ship in direction to the next point
13   var rotation = Quaternion.LookRotation(path.path_objs[currentWayPointId].position - transform.position);
14   transform.rotation = Quaternion.Slerp(transform.rotation, rotation, Time.deltaTime * rotationSpeed);
15   player.transform.rotation = ship.transform.rotation;
16
17   //if the distance between the ship and the point is less then reachDistance raise the currentwaypointid
18   if (distance <= reachDistance) {
19    currentWayPointId++;
20   }
21
22   //if the currentwaypoint is greater equal the size of the list
23   if (currentWayPointId >= path.path_objs.Count) {
24    //set ship to the last position
25    this.transform.position = path.path_objs[path.path_objs.Count-1].position;
26    //set playershipped status to true and make some other things
27    playerShipped = true;
28    cc.enabled = true;
29    playerOnShip = false;
30    player_mesh.SetActive(true);
31    player_text.SetActive(true);
32    follow.transform.parent = player.transform;
33    follow.transform.position = player.transform.position;
34
35    minidot.SetActive(true);
36    player.transform.position = new Vector3(path.path_objs[path.path_objs.Count - 1].position.x, 6.3f, path.pat
37
38    currentWayPointId = 0;
39   }
40  }
41 }
```

Listing 7: Methode moveShip

5.4.7 Datenimport aus JSON

Der Datenimport aus einer JSON-Datei wird genau zwei mal im Spiel verwendet. Die Standarts und die Texte der NPCs stehen in JSOn-Notation bereit. Dabei werden beide Dateien unterschiedlich ausgelesen und genutzt. Zuerst wird auf die Standarts eingegangen und anschleißend auf die Verarbeitung der Datei mit Text.

Standarts JSON

Die JSON für die Standarts ist ein wichtiger Bestandteil des Spieles. In dieser Datei befinden sich alle wichtigen Infos zu den Standarts und deren dazugehörigen Spiele. Diese Datei liegt Zentral auf einem Server und kann dort gewartet und gepflegt werden. Dadurch ist es möglich, zusätzliche Standarts und Spiele einzufügen, ohne das der Nutzer die App aktuallisieren muss. Nun wird der grobe Ablauf, gefolgt von einer detailierten Beschreibung, erklärt und beschrieben.

Damit der Nutzer die aktuelle Datei zur Verfügung hat, versucht das Spiel eine zu Beginn eine Verbindung zu dem Server aufzubauen. Gelingt dies, wird die Datei von Server geladen und auf dem Handy gespeichert. Nachdem die Datei gespeichert wurde, wird sie in das Spiel geladen. Dazu wird aus dem JSON eine Liste erstellt, welche überall im Spiel nutzbar ist.

Für das Laden und verarbeiten sind fünf Klassen verantwortlich. Im Folgenden wird nur deatiliert auf die Klasse WebJSONConfigReader eingegangen. Die anderen Klassen sind im Anhang abgebildet und werden nur erwähnt.

Die Klasse MappingStandardsToCourses ist die Datenhaltungsklasse. In dieser ist die Struktur der einzelnen Objekte geregelt. In der Klasse MappingStandardsToCoursesBean spiegelt die Struktur der JSON-Datei wieder.

Der WebJSONConfigReader wird in der Klasse LoadingScreen genutzt. Dort wird eine Instanz dieser Klasse erzeugt und es wird die Methode LoadSettings ausgeführt und in der Variable Settings gespeichert. Diese Methode ist in Listing 8 zu sehen.

```
1
2  public MappingStandardsToCourses LoadSettings() {
3
4   WWW www = new WWW(settingsURL);
5   while (!www.isDone) { }
6    string json;
7    string LocalFilePath = Application.persistentDataPath + "/v1.json";
8
9    if (string.IsNullOrEmpty(www.error)) {
10    json = www.text;
11    File.WriteAllText(LocalFilePath, json);
12   } else if (File.Exists(LocalFilePath)) {
13    json = File.ReadAllText(LocalFilePath);
14   } else {
15    json = "";
16    File.WriteAllText(LocalFilePath, json);
17   }
18   Debug.Log(json.ToString());
19   MappingStandardsToCoursesBean settingsBean = MappingStandardsToCoursesBean.CreateFromJSON(json);
20   string json2 = JsonUtility.ToJson(settingsBean);
21   Debug.Log(json2.ToString());
22
23   List<Classes> classes = new List<Classes>();
```

```
24   foreach (var classe in settingsBean.classes) {
25    List<Courses> courses = new List<Courses>();
26    foreach (var course in classe.courses) {
27     List<Standards> standards = new List<Standards>();
28     foreach (var standard in course.standards) {
29      List<Games> games = new List<Games>();
30      List<string> vor = new List<string>();
31      List<string> nach = new List<string>();
32      foreach (var game in standard.games) {
33       games.Add(new Games(game.id, game.name, game.url));
34      }
35      foreach (var vorbedingung in standard.vorbedingungen) {
36       vor.Add(vorbedingung);
37      }
38      foreach (var nachbedingung in standard.nachbedingungen) {
39       nach.Add(nachbedingung);
40      }
41      Games[] g = games.ToArray();
42      string[] v = vor.ToArray();
43      string[] n = nach.ToArray();
44      standards.Add(new Standards(standard.name, v, n, g));
45     }
46     Standards[] s = standards.ToArray();
47     courses.Add(new Courses(course.name, s));
48    }
49    Courses[] c = courses.ToArray();
50    classes.Add(new Classes(classe.name, c));
51   }
52   Classes[] cla = classes.ToArray();
53   return new MappingStandardsToCourses(cla);
54  }
```

Listing 8: Methode LoadSettings

In dieser wird zu Beginn ein neues WWW Objekt erstellt. Dieses baut eine Verbindung zum Server auf. Anschließend wird in Zeile 9 überprüft, ob es zu einem Fehler beim Verbindungsaufbau gekommen ist. Wenn es zu keinem Fehler gekommen ist, wird der Inhalt des WWW Objekts in eine Datei geschrieben und gespeichert. Für den Fall das ein Fehler aufgetretten ist und das WWW Objekt keine Verbindung zum Server aufgebaut werden kann, wird geprüft ob bereits eine ältere Datei vorhanden ist, wenn ja, wird der Text aus dieser Datei zum Erzeugen des Objektes genutzt. Anschlißend wird in Zeile 19 aus dem Text eine JSON Struktur erzeugt. Aus dieser wird anschließend in mehreren Schritten ein mehrdimensionaler Array erzeugt und zurückgegeben.

NPCText JSON

5.4.8 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 En-

de 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 gestarte, 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 dm 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 9: 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 abgegleicht. Sollte es hier einen Fehler geben, sendet der Server einen Error zurück, wenn die Hashes identisch sind, wird eine Verbindung zu der Datenbank aufgabenaut und ein Eintrag in der accounts Tabelle erstellt. Anschließend wird eine erfolgreich 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 validierug zu sehen.

Listing 10: Skript SendDataToServer.cs

5.4.9 Allgemein

5.5 Die Benutzerschnittstelle

In diesem Abschnitt dieses Kapitels wird auf die Implementierung der Benutzerschnittstelle eingegangen. Wie schon bereits im SRS erwähnt handelt es sich bei der Benutzerschnittstelle um die grafische Oberfläche der Android App. Dafür werden zunächst die Möglichkeiten von Unity vorgestellt, dann auf das Ergebnis der Umsetzung eingegangen und abschließend die Auswertung des Usability Tests behandelt.

5.5.1 Die Umsetzung

Um die ergonomie zu erhöhen kann der linke joystick i einem festgelegten bereich überall angelegt werden

Da es sich um KOmponenten handelt die immer da sein müssen wurde sich für das normale UI Screen Space - OVerlay ausgewählt

Abbildung 19: User Interface: Head-Up Display

Menu fenster bildet die erste Dialogebene nach der Startebene. Nach diesem gibt es noch eine weitere Ebene. dieser Dialog dient zum Abbrechen, warnt vor bevorstehenden Aktionen. Die maximale Dialogebene ist also 2.

5.5.2 Usability Test

Usability TEst faken? Kinder + Jugendliche –> Katalog mit Bewertung Summativer TEst: Nach der Entwicklung, fertiges Produkt vor Auslieferung umfangreiche Studie

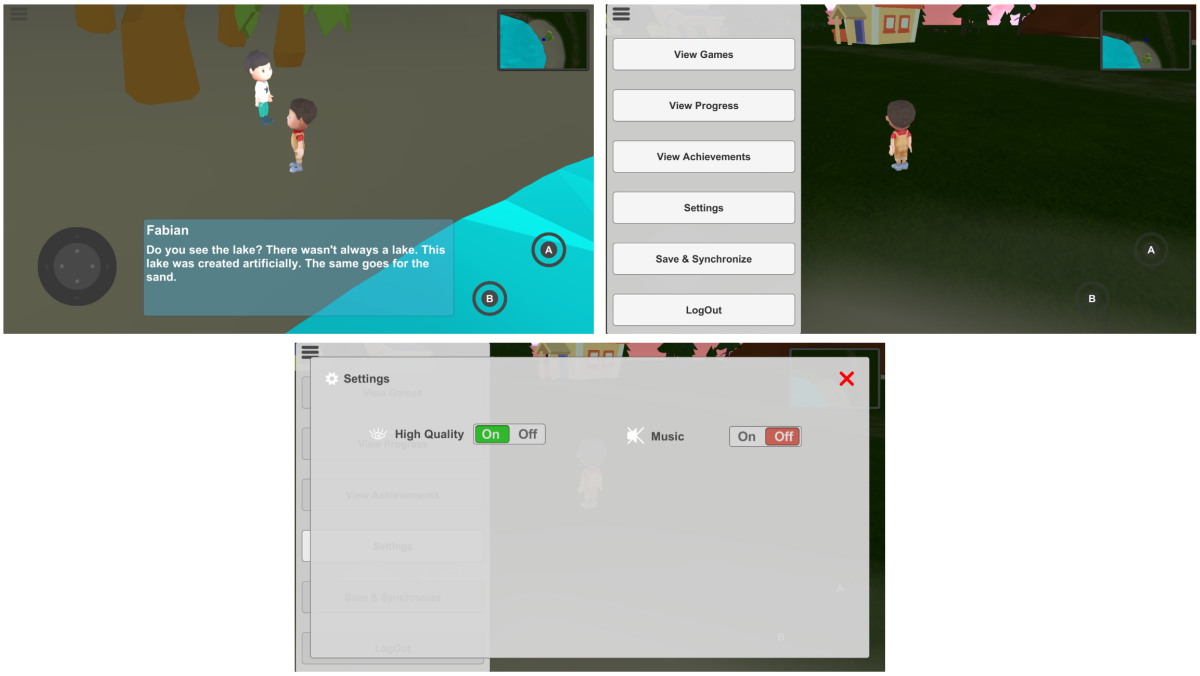

Abbildung 20: User Interfaces: Kommunikation, Menu und Einstellungen-Fenster

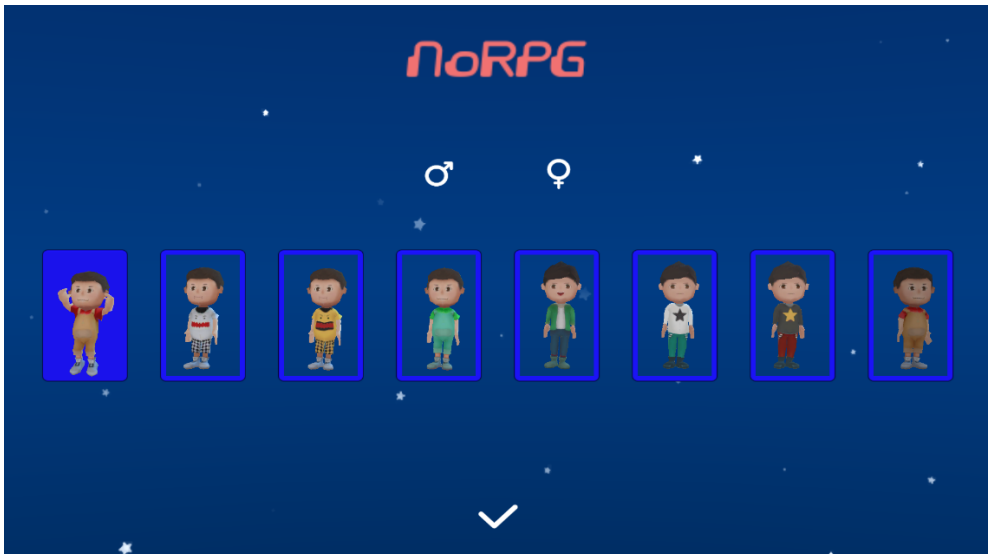

Abbildung 21: User Interface: Registrierung

Benutzerprofiele, Szenarien, Umfang, Zeitraum (3 Tage)

TEstutensilien: Prerelease version der APp und eigene Software Testart: Remotetest (vor und nachteile zu anderen)

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