



ulm university universität
uulm

Universität Ulm | 89069 Ulm | Germany

**Fakultät für
Ingenieurwissenschaften,
Informatik und Psychologie**
Institut für Datenbanken und
Informationssysteme

Konzeption und Realisierung einer mobilen Serious Games Anwendung zur Verbesserung der Geräuschlokalisierung für Tinnituspatienten

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Jana Bühler
jana.buehler@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

PD Dr. rer. nat. Winfried Schlee und Dr. Rüdiger Pryss

Abgabedatum:

5.11.2018

Fassung 29. Oktober 2018

© 2018 Jana Bühler

This work is licensed under the Creative Commons Attribution 4.0 International (CC BY 4.0) License. To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Satz: PDF- \LaTeX 2 ϵ

Abstract

Etwa 9 Prozent der Menschen in Deutschland leiden zumindest gelegentlich an Tinnitus. In der Literatur werden verschiedene Behandlungsmaßnahmen diskutiert. Neben den klassischen Therapieansätzen stellen mobile Health-Anwendungen neue Möglichkeiten dar. Der emotionale Zustand des Patienten hat einen Einfluss auf die Beschwerden. Um negativen Emotionen entgegenzuwirken, soll in der vorliegenden Arbeit eine mobile Serious Games Anwendung entwickelt werden. Diese trainiert das Richtungshören durch Geräuschlokalisierungen und ermöglicht so das Trainieren der selektiven Wahrnehmung für Tinnituspatienten. In der entwickelten Anwendung befindet sich der Spieler in einem Geisterschloss und soll auftauchende Geister fotografieren, um diese verschwinden zu lassen. Die Geister sind nicht sichtbar. Der Spieler muss durch aufmerksames Hören sein Smartphone in die Richtung drehen, aus der das Geräusch kommt. Durch Score-Punkte wird die Motivation hochgehalten. Das Ergebnis dieser Arbeit ist eine funktionierende Serious Games Anwendung für Android-Geräte, die nach zukünftigen Erweiterungen im Google Play Store oder einem anderen App Store bereitgestellt werden kann.

Inhaltsverzeichnis

Abstract	iii
Abbildungsverzeichnis	vi
Tabellenverzeichnis	viii
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Zielsetzung	2
1.3 Aufbau der Arbeit	3
2 Grundlagen	4
2.1 Serious Games	4
2.2 Tinnitus	5
3 Vorgehensweise und Anforderungen an die Anwendung	8
3.1 Agiles Vorgehensmodell	8
3.2 Pivotal Tracker	10
3.3 Produktvision	11
3.4 User Stories und Akzeptanzkriterien	11
4 Konzeption	14
4.1 Spielkonzept	14
4.1.1 Spielidee	14
4.1.2 Lerninhalt	15
4.1.3 Zielgruppe und Genre	15
4.1.4 Geräuschlokalisierung	16
4.2 Mockups	17
4.3 Navigationskonzept	19
4.4 Werkzeuge	21
4.4.1 Unity	21

4.4.2	Android Studio	21
4.4.3	Blender	22
5	Realisierung und Validierung	23
5.1	Architektur	23
5.2	Ausgewählte Implementierungsdetails	27
5.2.1	Erstnutzung der Anwendung	27
5.2.2	Spielernamen wählen	28
5.2.3	Kopfhöreranschluss	28
5.2.4	Audio	29
5.2.5	Geister	30
5.2.6	Auftauchen der Geister	32
5.2.7	Gyroskop	33
5.2.8	Fotografieren	34
5.2.9	Highscore	35
5.2.10	Ahnengalerie	36
5.2.11	Zusammenführung Unity- und Android Studio Projekt	36
5.3	Vorstellung der Anwendung	38
5.4	Erfüllung der Akzeptanzkriterien	43
6	Fazit	45
6.1	Zusammenfassung	45
6.2	Ausblick	46
	Literaturverzeichnis	47

Abbildungsverzeichnis

2.1	Maskierungstherapie	6
3.1	Verwendeter Ablauf	9
3.2	Pivotal Tracker	10
4.1	Spielprinzip	16
4.2	Startbildschirm	17
4.3	Beispiel einer Spielszene	17
4.4	Highscore	18
4.5	Erweitertes Menü	18
4.6	Ahnengalerie	18
4.7	Geister	19
4.8	Start des Spiels	19
4.9	Bestenliste ansehen	20
4.10	Spielernamen ändern	20
5.1	Komponenten der Anwendung	23
5.2	Klassendiagramm Highscore	24
5.3	Klassendiagramm App-Modul	25
5.4	Aufbau des GhostGames in Unity	26
5.5	Wahrnehmung einer Audio-Quelle [13]	30
5.6	Entstehung des Geistes	31

ABBILDUNGSVERZEICHNIS

5.7 Accessoires	31
5.8 Konzept: Auftauchen der Geister	32
5.9 Erster Start	38
5.10 Startbildschirm	39
5.11 Spielmodus	40
5.12 Räume aus dem Schloss	40
5.13 Geister	41
5.14 Highscore	41
5.15 Ahnengalerie	42
5.16 Spielername ändern	42
5.17 Informationen über die GhostApp	43

Tabellenverzeichnis

2.1	Schweregradeinteilung beim Tinnitus [2]	5
5.1	Erfüllung der Akzeptanzkriterien	44

1 Einleitung

1.1 Motivation und Problemstellung

Etwa 9 Prozent der Menschen in Deutschland leiden zumindest gelegentlich an Tinnitus [29]. Man spricht von Tinnitus, wenn der Betroffene „Geräuschsensationen wahr[nimmt], ohne dass eine objektivierbare externe Schallquelle vorhanden ist“ [2]. Es gibt Patienten, die sich an die Ohrgeräusche gewöhnen, aber auch Betroffene, die stark in ihrem Alltag eingeschränkt sind [25]. Aus medizinischer Sicht gibt es körperliche und psychosomatische Ursachen für Tinnitus [11]. Laut Hamann sind die „häufigsten körperlichen Ursachen [...] Schädigungen des Innenohrs, wie sie zum Beispiel durch extreme Lärmeinwirkungen entstehen.“ Auch Stress und Überforderung können zu Ohrgeräuschen führen.

Eine wichtige Frage ist, wie man den Betroffenen am besten helfen kann, da es keine effektive medikamentöse Therapie gibt [2]. In der Literatur sind verschiedene Therapieformen und Behandlungsmaßnahmen beschrieben [2, 11, 12].

In letzter Zeit hat sich der mobile Crowdsensing-Ansatz etabliert [21]. Hierbei können große Datenmengen zu relativ geringen Kosten gesammelt werden [21]. In [23] wird eine Crowdsensing-Plattform vorgestellt, die einen mobilen Patientenfeedbackdienst als Komponente enthält. Im Vergleich zu traditionellen Forschungsmethoden können die Daten mit Hilfe von mobilen Health-Technologien prospektiv erfasst werden [19, 20]. Der Vorteil von mobilen Health-Anwendungen ist, dass diese leicht im Alltag anwendbar sind [22] und die Daten im Vergleich zu klinischen Studien in natürlicher Umgebung erfasst werden [24]. Mit traditionellen Forschungsmethoden ist dies schwierig durchzuführen, daher wurde eine mobile Anwendung, namens 'TrackYourTinnitus', entwickelt [26]. Mit Hilfe dieser Anwendung kann der Patient die Tinnituswahrnehmung in Alltagssituationen aufnehmen. Die Daten können dann Aussagen über die Schwankungen der Tinnituswahrnehmung machen [18].

Ein Problem bei der Tinnitusbehandlung ist, dass der emotionale Zustand des Patienten einen Einfluss darauf hat, ob das Wahrnehmen des Tinnitussignals zu Tinnitusbeschwerden führt [18]. Negative Emotionen lenken die Aufmerksamkeit des

Patienten auf das Tinnitusignal, sodass dieses verstärkt wahrgenommen wird [2]. Daher sollte versucht werden, eine Methode bei der Behandlung einzusetzen, die die Motivation des Patienten hoch hält.

In [35] wird eine spielbasierte Trainingsmethode für die Behandlung von Tinnitus vorgestellt. In dem Spiel wird die Lokalisierung von Geräuschen und die selektive Wahrnehmung trainiert. In einer Studie haben 8 Teilnehmer das Spiel über 20 Tage jeweils 30 Minuten pro Tag genutzt. Das Ergebnis der Studie war, dass das Spiel eine gute Alternative zur traditionellen Tinnitustherapie darstellt. Auch kann das Spiel verwendet werden, um das auditorische System besser verstehen zu lernen.

Digitale Spiele, die nicht nur der reinen Unterhaltung dienen, sondern einen weiteren Zweck erfüllen, werden als Serious Games bezeichnet [4]. Im Bereich e-Health und mobile Health gewinnt der Einsatz von Serious Games Anwendungen immer mehr an Bedeutung [33].

In [25] wird erstmalig ein mobiles Serious Games Spielkonzept vorgestellt, das für die Behandlung von Tinnitus eingesetzt werden kann. Es beruht auf dem Ansatz, dem Betroffenen Tiergeräusche in einem dreidimensionalen Raum an einer zufälligen Position und einer entfernungsabhängigen Intensität vorzuspielen. Während des Spielens werden verschiedene Daten gesammelt, insbesondere die Zeit, die ein Spieler benötigt, das Tier zu finden und zu fotografieren, sowie die Winkelabweichung vom optimalen Treffpunkt. Die Serious Games Anwendung wurde für drei verschiedene mobile Betriebssysteme realisiert. In einer Benutzerstudie mit 24 Personen, von denen 2 von Tinnitus betroffen waren, wurde die Anwendung evaluiert. Obwohl es punktuelle Unterschiede zwischen den Betriebssystemen gab, ist die Nutzung der App auf allen mobilen Plattformen möglich und hilfreich.

Die vorliegende Arbeit soll zentrale Konzepte von [25] aufgreifen und diese in einem neu zu konzipierenden Spiel anwenden. Dies betrifft vor allem das Verwenden von 3D-Geräuschen und das zufällige Positionieren im virtuellen 3D-Raum.

1.2 Zielsetzung

Im Rahmen dieser Arbeit soll eine mobile Serious Games Anwendung entwickelt werden, mit der Tinnituspatienten trainieren, den Fokus vom Tinnitusignal auf andere Geräusche zu lenken. Durch die Kombination aus Unterhaltung und ernsthaftem Zweck kann laut Schickler et al. die Motivation erhöht und damit der Behandlungserfolg verbessert werden [25]. Die zu entwickelnde mobile Anwendung soll für

Android-Geräte entwickelt werden. Das Spiel soll unterhaltsam und für alle Altersgruppen geeignet sein. Um die selektive Aufmerksamkeit trainieren zu können, ist die Integration von 3D-Geräuschen von großer Bedeutung.

1.3 Aufbau der Arbeit

Nach diesem einleitenden Kapitel werden in Kapitel 2 zunächst die grundlegenden Begriffe Serious Games und Tinnitus erläutert. In Kapitel 3 wird beschrieben, nach welchem Vorgehensmodell die Serious Games Anwendung entwickelt wurde. Anschließend werden die Produktvision und die Anforderungen in Form von User Stories und Akzeptanzkriterien festgelegt. Kapitel 4 beschreibt die zugrunde liegenden Konzepte, welche das Spielkonzept, die Mockups, das Navigationskonzept und die verwendeten Werkzeuge beinhalten. In Kapitel 5 werden die Anwendungsarchitektur, technisch interessante Implementierungsdetails, sowie die entstandene Anwendung vorgestellt. Im Anschluss daran wird dargestellt, welche Akzeptanzkriterien erfüllt wurden. Das letzte Kapitel enthält das Fazit mit einer Zusammenfassung der Arbeit und einem Ausblick.

2 Grundlagen

2.1 Serious Games

Der Begriff Serious Games hat sich in den letzten Jahren weit verbreitet. Trotzdem gibt es derzeit keine einheitliche Definition in der Literatur [33]. Nach [4] versteht man unter Serious Games digitale Spiele, die unterhalten sollen und mindestens ein weiteres Ziel verfolgen. Dieses kann beispielsweise aus dem Bereich Lernen oder Gesundheit stammen. In anderen Definitionen wird ein Spiel durch die Intention des Spielers zu einem Serious Game [4]. So wird das Spiel zum Serious Game, wenn der Spieler die Intention hat, das Spiel nicht nur zur Unterhaltung zu spielen, sondern zum Trainieren. Andere verlangen, dass das Spiel freiwillig von dem Spieler in seiner Freizeit gespielt wird und nicht Teil des Bildungsumfeldes ist [4]. Eine weitere Definition liefert Göbel in [30]: „Es muss in erster Linie ein Spiel sein. Es reicht nicht, dass es eine reine Trainings- und Simulationssoftware ist, sondern es muss ein Spiel sein, es muss Spaß machen und zusätzlich diesen übergeordneten Zweck erfüllen.“ Der letzte Ansatz wird auch in dieser Arbeit verfolgt.

Serious Games können aus den Bereichen Training und Simulation, Pädagogik, Gesundheit und Gesellschaft stammen [4]. Anwendungen aus dem Bereich Gesundheit werden insbesondere für Präventions- und Rehabilitationsmaßnahmen eingesetzt [34]. Als Beispiele werden Ernährung, Energieverbrauch, Krebstherapie und neurologische Erkrankungen genannt. Wiemeyer begründet den Einsatz von Serious Games damit, dass therapeutische Maßnahmen über einen längeren Zeitraum durchgeführt werden müssen und die spielerischen Elemente dazu dienen, einen langfristigen Erfolg zu gewährleisten [34]. Für die verschiedenen Bereiche werden Experten aus den Disziplinen Informatik, Kunst und Design, Psychologie sowie Pädagogik zur Entwicklung der Spiele benötigt [4].

2.2 Tinnitus

Tinnitus ist ein Symptom, bei dem der Betroffene Ohrgeräusche wahrnimmt [11]. Hamann unterscheidet zwischen akutem Tinnitus, der bis zu drei Monate anhalten kann und chronischem Tinnitus, sofern dieser mindestens ein Jahr auftritt. Die Ohrgeräusche sind in der Regel nur von dem Betroffenen wahrnehmbar. Dann spricht man von einem subjektiven Tinnitus [2]. Lässt sich der Schall der Geräusche messen, so spricht man von einem objektiven Tinnitus [2]. Hierbei gehen Biesinger und Iro davon aus, dass eine eindeutige Ursache vorliegt. Das Ohrgeräusch wird von Hamann in tonal und nicht tonal unterschieden [11]. Handelt es sich um ein Pfeifen, Klingeln, Summen oder Zirpen, liegt ein tonaler Tinnitus vor. Andernfalls, wenn der Betroffene ein Rauschen, Brummen oder Surren wahrnimmt, redet man von einem nicht tonalen Tinnitus [11]. Die Ohrgeräusche können unterschiedlich intensiv auftreten und werden in [2] wie in Tabelle 2.1 klassifiziert:

Einteilung	Bezeichnung
Grad 1	Der Tinnitus ist gut kompensiert, es besteht kein Leidensdruck
Grad 2	Der Tinnitus wirkt zeitweise störend in bestimmten Situationen und bei bestimmten Belastungen
Grad 3	Der Tinnitus führt zu einer dauernden Beeinträchtigung im privaten und/oder beruflichen Bereich. Es treten Störungen im emotionalen, kognitiven und körperlichem Bereich auf. Es besteht noch Arbeitsfähigkeit
Grad 4	Der Tinnitus führt zu einer völligen Dekompensation im privaten Bereich, Erwerbsunfähigkeit

Tabelle 2.1: Schweregradeinteilung beim Tinnitus [2]

Hamann beschreibt körperliche und psychosomatische Ursachen für Tinnitus [11]. Der häufigste körperliche Grund seien Schädigungen des Innenohrs, können aber auch durch viele weitere Erkrankungen hervorgerufen werden. Eine psychosomatische Ursache liegt laut Hamann vor, wenn der Körper Alarmsignale in Form von Tinnitus sendet. Diese können hauptsächlich durch ein Trauma, Stress oder Überforderung ausgelöst werden.

Emotionen spielen beim Tinnitus eine große Rolle. Durch negative Emotionen schenkt der Patient dem Tinnitusignal starke Aufmerksamkeit, wodurch das limbische System aktiviert wird [2]. Dieses ist für die Erstellung und Verarbeitung von Erinnerungen und die Bewertung von Emotionen zuständig [11]. Dies führt zu einer Spirale. Daher ist es wichtig, dass der Patient keine negativen Emotionen im Bezug auf seinen Tinnitus entwickelt und möglichst wenig Aufmerksamkeit auf das Ohrgeräusch legt. Dies ist bei der Entwicklung der Serious Games Anwendung besonders zu beachten.

Tinnitus kann man nicht heilen und es gibt keine effektive medikamentöse Therapie, sodass auf andere Methoden zurückgegriffen werden muss [2]. Neben den klassischen Therapieansätzen, wie zum Beispiel die ambulante Tinnitustherapie oder Psychotherapie hat sich vor allem die Musiktherapie etabliert [12]. Hierbei gibt es zwei Ansätze, die von Biesinger und Iro beschrieben werden. In der Maskierungstherapie hört sich der Patient ein externes Geräusch an (siehe Abbildung 2.1 a). Das Geräusch wird in der Frequenz und Höhe so angepasst, dass das Tinnitusignal verkleinert wird. Wenn der Tinnitus durch die Geräuschkulisse nicht mehr zu hören ist, ist der Tinnitus vollständig maskiert (siehe Abbildung 2.1 c), ansonsten spricht man von einer Teilmaskierung (siehe Abbildung 2.1 b). Ziel ist es hierbei, dass das Tinnitusignal nicht wahrgenommen wird und der Patient das externe Geräusch leichter erträgt und verdrängen kann.

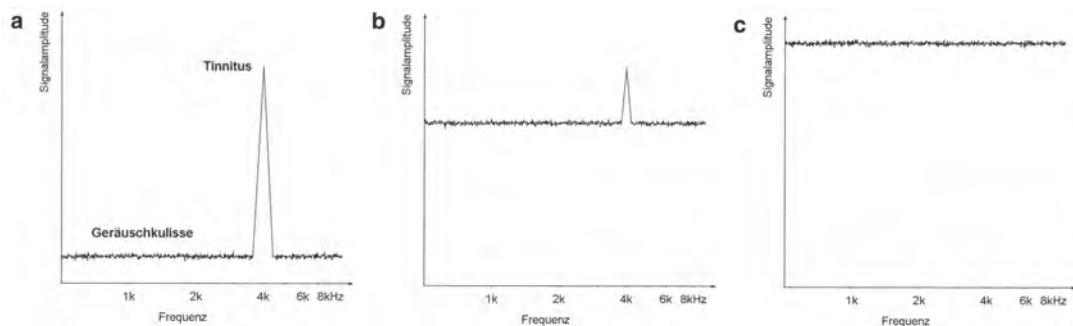


Abbildung 2.1: Maskierungstherapie

Der zweite Therapieansatz ist die "tinnitus retraining therapy", kurz TRT. Es wird ein Geräuschgenerator eingesetzt, der die Separierung des Tinnitussignals erschweren soll. Hierbei wird ein möglichst breitbandiges, gering überschwelliges und stabiles Hintergrundgeräusch verwendet. Dieser Ansatz geht davon aus, dass eine vollständige Maskierung des Tinnitussignals kontraproduktiv sei und legt den Schwerpunkt auf die Gewöhnung an das Signal. Die TRT wird sehr häufig eingesetzt, da eine erfolgreiche Therapie eine dauerhafte Habituation ermöglicht.

3 Vorgehensweise und Anforderungen an die Anwendung

3.1 Agiles Vorgehensmodell

Zur Entwicklung der Serious Games Anwendung wurde ein agiler Entwicklungsansatz nach Scrum gewählt. „Scrum ist ein Rahmenwerk, innerhalb dessen Menschen komplexe, adaptive Aufgabenstellungen angehen können [...]“ [27]. Vorteilhaft dabei ist, dass Scrum einfach zu verstehen ist und eine iterative, inkrementelle Vorgehensweise unterstützt. Das heißt, dass das Produkt schrittweise in kleinen Aufbaustufen erstellt wird und möglichst früh eine lauffähige Version entsteht. Des Weiteren können die Erfahrungen aus früheren Schritten genutzt werden [5].

Im Manifest für Agile Softwareentwicklung [1] werden Werte sowie deren Gewichtung als Grundlage festgelegt, die auch bei der Entwicklung mit Scrum Verwendung finden:

- Individuen und Interaktionen mehr als Prozesse und Werkzeuge
- Funktionierende Software mehr als umfangreiche Dokumentation
- Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
- Reagieren auf Veränderung mehr als das Befolgen eines Plans

Durch das agile Vorgehen soll eine schnell lauffähige Software entwickelt werden, die einen geringen bürokratischen Aufwand aufweist. Es wird eine erfahrungsbezogene Projektkontrolle gefordert. Durch das adaptive Planen hat man eine hohe Flexibilität im Projekt.

Der Scrum-Prozess hat einen bestimmten Ablauf. Angelehnt an diesen wurde der in Abbildung 3.1 dargestellte Ablauf verwendet.

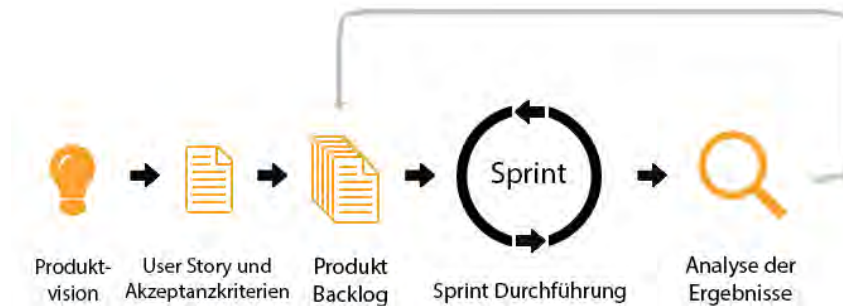


Abbildung 3.1: Verwendeter Ablauf

Zu Beginn des Prozesses wird die Produktvision erstellt. Mit dessen Hilfe wird die Motivation für das Projekt und die angestrebten Projektziele beschrieben [5]. Nach Goll legt diese die Richtung, die Grundlage für das Projekt ist, fest. Hier wird unter anderem definiert, wer die Zielgruppe des Produktes ist, welche Bedürfnisse erfüllt werden sollen und welche Eigenschaften am Wichtigsten zur Erfüllung der Bedürfnisse sind [5]. Als nächstes werden User Stories erstellt. Eine User Story beschreibt eine funktionale Anforderung, die für den Benutzer wichtig ist und einen bestimmten Zweck erfüllt. Um eine User Story zu schreiben, wird folgendes Satzschema eingehalten:

„Als [Rolle] möchte ich __, um __.“

Ein konkretes Beispiel ist: *„Als Benutzer möchte ich in einem Telefonbuch suchen können, um eine bestimmte Telefonnummer zu finden.“*

Um verifizieren zu können, ob eine User Story vollständig umgesetzt wurde, muss diese genau formulierte Bedingungen enthalten. Die Bedingungen werden als Akzeptanzkriterien festgelegt. Alle User Stories werden in dem Produkt Backlog zusammengefasst. Nach [5] wird der Entwicklungsprozess in Iterationen, die man Sprints nennt, aufgeteilt. Für jeden Sprint werden User Stories aus dem Produkt Backlog ausgewählt, die in dieser Iteration entwickelt werden sollen. Ein Sprint geht in der Regel wenige Wochen. Nach dem Sprint wird dieser analysiert. So kann der Projektablauf während der Laufzeit angepasst und optimiert werden. Da hier der Gesamtüberblick durch eine klare Projektübersicht nicht gefördert wird, erfordert dies ein hohes Maß an Disziplin, um ein erfolgreiches Projekt abzuliefern. Um dies besser bewältigen zu können, wurde in dieser Arbeit der Pivotal Tracker [15] verwendet.

3.2 Pivotal Tracker

Der Pivotal Tracker [15] ist ein Projektplanungs-Tool, das vor allem für agile Projekte geeignet ist. Es unterstützt die Formulierung und Visualisierung der zu realisierenden Aufgaben [16]. Hierzu werden virtuelle Karten erstellt, die durch den Projekt-Workflow bewegt werden. Dadurch hat man zu jedem Zeitpunkt eine nachvollziehbare Projektübersicht. Eine virtuelle Karte stellt eine zu erledigende Aufgabe präzise dar. Es wird der Aufwand geschätzt und die Aufgabe priorisiert. Wurden alle Aufgaben erstellt, können diese bestimmten Iterationen zugewiesen werden. Dies erleichtert die Planung und Anpassung der weiteren Arbeitsschritte.

In Abbildung 3.2 ist das Tool mit Beispielkarten zu sehen. Hier wird der Backlog Icebox genannt. In der rechten Box sind Karten der aktuellen Iteration in verschiedenen Zuständen zu sehen. Sie sind einem bestimmten Zeitraum zur Bearbeitung zugewiesen.

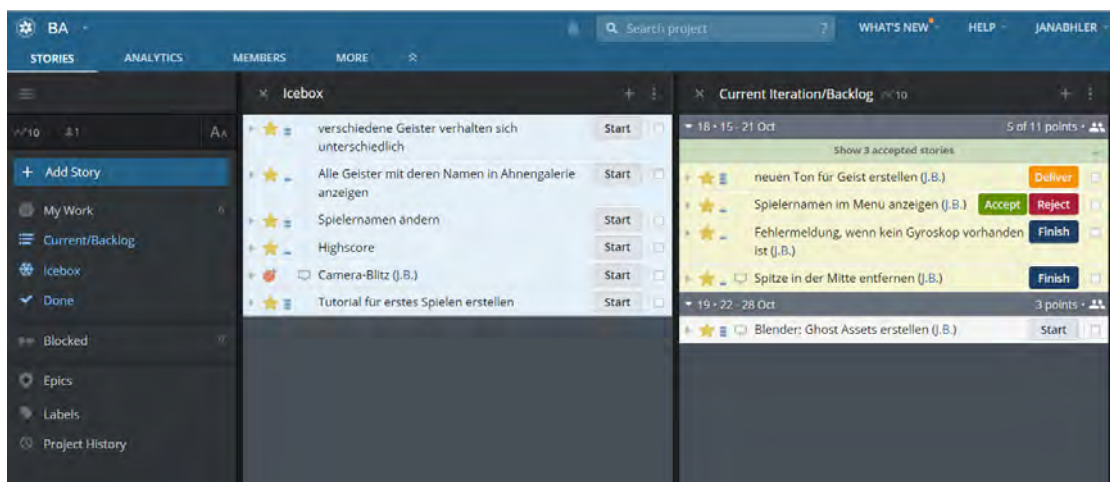


Abbildung 3.2: Pivotal Tracker

3.3 Produktvision

Gemäß dem agilen Vorgehen nach Scrum startet ein Projekt zunächst mit der Beschreibung der Produktvision. Das zugrunde liegende Szenario beruht auf der Geräuschlokalisierung sowie auf der Konzentration auf ein bestimmtes Geräusch. Da der Spaß eine wichtige Rolle spielt, soll eine Serious Games Anwendung entwickelt werden, die ein Geisterschloss als Basis in einem virtuellen 3D-Raum enthält. Nachfolgend wird die Produktvision für die zu entwickelnde App angegeben.

Mit Hilfe des Spiels soll das räumliche Hörvermögen verbessert werden. Inhaltliches Ziel des Spiels ist es, das Schloss, in dem sich der Spieler befindet, von Geistern zu befreien. Diese tauchen im Schloss um den Spieler herum auf und können mithilfe einer Kamera fotografiert werden. Durch den Blitz der Kamera verschwindet der Geist. Das Spiel macht den Spielern aller Altersklassen Spaß und stärkt unbewusst das räumliche Hörvermögen. Besonders hilfreich ist das Spiel für Tinnituspatienten zum Trainieren der selektiven Aufmerksamkeit. Das Spiel soll so gestaltet sein, dass der Spieler sich wohlfühlt und eine hohe Motivation aufrecht erhalten wird.

3.4 User Stories und Akzeptanzkriterien

Um konkret zu spezifizieren, was der Benutzer mit Hilfe der Anwendung machen kann und warum er dies tun möchte, wurden User Stories (US) erstellt. Die Akzeptanzkriterien (A) beschreiben, wann die Funktionalität erfüllt ist und werden zu jeder User Story dargestellt.

US1: Als Benutzer möchte ich die App starten können, um das Spiel zu spielen.

A1: Der Benutzer klickt auf das Icon der App auf seinem Android-Smartphone.

A2: Öffnet der Benutzer das Spiel zum ersten Mal, wird ein Willkommensbildschirm angezeigt, auf dem der Benutzer einen Spielernamen wählen kann. Ansonsten wird ein Startbildschirm angezeigt.

A3: Auf dem Startbildschirm werden die Buttons zum Menüausklappen, Starten des Spiels und Anzeigen des Highscores dargestellt.

A4: Durch Drücken des Buttons 'Spielen' kann das Spiel gestartet werden.

- US2: Als Benutzer möchte ich meinen Spielernamen ändern können, um meine Spielergebnisse unter meinem Namen abzuspeichern.
- A1: Der Benutzer klappt das Menü aus und klickt auf 'Geisterjägername wählen'.
 - A2: Der Benutzer gibt einen Spielernamen ein.
 - A3: Das System speichert die Angaben auf dem Smartphone.
 - A4: Nach dem Speichern wird der Startbildschirm mit kurzer Begrüßung angezeigt.
- US3: Als Benutzer möchte ich meine besten Spielergebnisse speichern und ansehen können, um mein aktuelles Spiel mit Bisherigen vergleichen zu können.
- A1: Der Benutzer befindet sich auf dem Startbildschirm und klickt auf den Button 'Highscore'.
 - A2: Das System zeigt die besten fünf Ergebnisse in einer Liste an.
 - A3: Nach dem Spielende werden die erreichten Punkte im System gespeichert.
- US4: Als Benutzer möchte ich das Spiel spielen können, um mein Orientierungshören zu verbessern.
- A1: Der Startbildschirm wird angezeigt und der Benutzer drückt auf 'Spielen'.
 - A2: Das Spiel wird initialisiert und der Score wird auf Null gesetzt.
 - A3: Es wird ein Schloss und der Score angezeigt.
 - A4: Der Benutzer hört einen Geist und versucht sich ihm zuzuwenden, indem er das Smartphone in die Richtung des Geräusches dreht.
 - A5: Wenn der Benutzer der Meinung ist, dass er den Geist gefunden hat, tippt er auf den Bildschirm des Smartphones und es wird ein Blitz als Feedback angezeigt.
 - A6: Wenn der Benutzer richtig lag, erscheint der gefundene Geist für einige Sekunden und der Spieler bekommt anteilig Score-Punkte, je nach dem wie weit der Treffpunkt von der optimalen Position entfernt war.
 - A7: Wenn der Benutzer falsch lag, wird ihm dies angezeigt und er bekommt keine Score-Punkte.

- A8: Nachdem der gefundene Geist verschwindet, muss der nächste Geist gefunden werden.
- A9: Ein Spieler hat nur eine begrenzte Zeit, den Geist zu finden.
- A10: Das aktuelle Spiel ist verloren, wenn die Zeit abgelaufen ist. Als Rückmeldung wird dem Spieler ein Hinweistext angezeigt und der Startbildschirm wird aufgerufen.
- US5: Als Benutzer möchte ich das Spiel in einem Tutorial erklärt bekommen, um das Spielprinzip schneller zu verstehen.
- A1: Der Startbildschirm wird angezeigt und der Benutzer drückt auf „Spielen“.
- A2: Das Spiel wird initialisiert und der Raum wird angezeigt.
- A3: Dem Spieler werden Hilfstexte angezeigt, um das Spielprinzip kennen zu lernen.
- A4: Im Gegensatz zum normalen Spielmodus sind die Geister sichtbar, um es dem Spieler leichter zu machen.
- A5: Die Bedienung des Spiels funktioniert identisch zum normalen Spielmodus.
- US6: Als Benutzer möchte ich eine Übersicht über die Geister erhalten, damit ich weiß, welche Geister im Spiel gefunden werden sollen.
- A1: Der Benutzer befindet sich auf dem Startbildschirm, öffnet das seitliche Menü und klickt auf den Menüpunkt 'Ahnengalerie'.
- A2: Das System zeigt die Geister zusammen mit einer Beschreibung in einer Liste an.
- US7: In der App soll sichergestellt werden, dass das Spiel nur gestartet werden kann, wenn die Systemvoraussetzungen erfüllt sind.
- A1: Das System prüft beim Starten des Spiels, ob Kopfhörer an dem Smartphone angeschlossen sind.
- A2: Falls dies der Fall ist, wird das Spiel initialisiert und gestartet. Andernfalls wird ein Warnhinweis angezeigt und das Spiel wird nicht gestartet.
- A3: Anschließend überprüft das System, ob das Smartphone ein Gyroskop hat. Das Spiel soll nicht gestartet werden, wenn kein Gyroskop vorhanden ist.

4 Konzeption

4.1 Spielkonzept

4.1.1 Spielidee

Das zu entwickelnde Spiel muss so gestaltet sein, dass ein virtueller 3D-Raum Verwendung findet und ein zufällig platzierter 3D-Ton natürlich wirkt. Da eine breite Zielgruppe angesprochen werden soll, wurde eine Spielwelt gesucht, die von möglichst vielen Personen das Interesse weckt. Die Wahl fiel schließlich auf ein Geisterschloss. Hier können Geister an verschiedenen Positionen auftauchen.

Das Thema Geister und Geisterschloss ist bei vielen Personen beliebt. So planten 2017 rund 6% der Erwachsenen in Deutschland, die sich an Halloween verkleiden, sich als Gespenst zu verkleiden [28]. Das entspricht der fünft meist gegebenen Antwort.

Das festgelegte Thema ist geeignet, da es emotional positiv belegt ist und so als Motivationsverstärker wirken kann, sodass Benutzer das Spiel auch regelmäßig spielen wollen.

Der Spieler befindet sich in einem Geisterschloss und muss die auftauchenden Geister vertreiben. Diese tauchen nacheinander in dem Raum, in dem sich der Spieler befindet, auf - sind jedoch unsichtbar und müssen durch ihre Geräusche entdeckt werden. Um einen Geist vertreiben zu können, muss der Spieler diesen fotografieren, indem er auf das Display seines Smartphones tippt. Das Smartphone wird vor dem Körper in den Händen gehalten und erkennt Bewegungen des Spielers, sodass dieser sich durch Drehungen seines Körpers im 3D-Raum drehen kann. So kann sich der Spieler dem Geist zuwenden. Ziel ist es, dass der Spieler möglichst genau die Position des Geistes hört und sich diesem zuwendet. Meint der Spieler die korrekte Position gefunden zu haben, fotografiert er den Geist und erhält ein visuelles Feedback, ob und wie genau der Geist getroffen wurde. Beim Fotografieren wird der Winkel zwischen dem Geist und dem optimalen Treffpunkt in

der Mitte des Bildschirms gemessen. Je besser der Geist getroffen wurde - also je geringer der Winkel ist - desto mehr Punkte bekommt der Spieler. Außerdem muss jeder Geist in einer bestimmten Zeit entdeckt werden, sonst verliert der Spieler. Hat der Spieler alle Geister in einem Raum vertrieben, so wird der Raum gewechselt und neue Geister erscheinen.

Die verschiedenen Räume im Schloss können als Level interpretiert werden. Sobald alle Geister in einem Raum entdeckt wurden, ist das Level abgeschlossen. Jeder Raum wird schwieriger, das bedeutet, die Zeit zwischen Finden des Geistes und Auftauchen des Nächsten wird kürzer. Außerdem müssen die Geister schneller entdeckt werden.

4.1.2 Lerninhalt

Neben der Unterhaltung soll ein weiteres Ziel in der Serious Games Anwendung verfolgt werden. In dem zu entwickelnden Spiel soll die Lokalisierung von Geräuschen trainiert werden. Der Spieler muss hören, aus welcher Richtung das Geräusch kommt und sich dann richtig drehen. Daher ist hier entscheidend, dem Spieler eine Rückmeldung zu geben, wie gut er die Position des Geistes getroffen hat. Dies erfolgt über visuelles Feedback und Score-Punkte.

Die Serious Games Anwendung kann insbesondere auch verwendet werden, damit Tinnituspatienten die Fokussierung auf ein bestimmtes Geräusch trainieren.

4.1.3 Zielgruppe und Genre

Das Spiel ist für Menschen aller Altersgruppen geeignet. Vor allem Tinnituspatienten bilden die Zielgruppe, jedoch auch Personengruppen, die die Lokalisierung von Geräuschen trainieren wollen. Des Weiteren soll das Spiel verwendet werden können, wenn es nur der Unterhaltung dienen soll.

Bei dem Spiel handelt es sich um ein Serious Game für den Gesundheitsbereich. Es ist ein endlos laufendes Spiel, in dem der Spieler mit sich selbst konkurriert, um seinen bisherigen Rekord zu übertreffen und oben auf der Bestenliste zu stehen. Das Spiel ist besonders gut für Smartphones geeignet, da das Spiel ausschließlich durch Tippen auf den Bildschirm und Drehen des Smartphones gesteuert wird. Dies bietet eine einfache und intuitive Steuerung.

4.1.4 Geräuschlokalisierung

Ein zentraler Punkt der Anwendung ist die Geräuschlokalisierung. Um ein räumliches Hören zu ermöglichen, muss dies geeignet in der Anwendung realisiert werden. Als Grundlage für 3D-Audio soll nach [25] binaurales Hören verwendet werden. Das Gehirn analysiert Zeit- und Intensitätsschwankungen, die durch die seitliche Positionierung der Ohren am Kopf verursacht werden [25].

Da Schallwellen nicht gleichzeitig an beiden Ohren eintreffen, können tiefe Töne in horizontaler Richtung bestimmt werden [6]. Je nachdem, wie weit sich die Quelle auf einer Seite des Kopfes befindet, brauchen die Schallwellen länger zum anderen Ohr und der Zeitunterschied ist größer. Höhere Töne können durch die Lautstärke in der horizontalen Position bestimmt werden [6]. Je nach dem, aus welcher vertikalen Richtung ein Ton kommt, reflektiert dieser unterschiedlich in unserem Ohr.

Problematisch ist, wenn der Spieler nicht unterscheiden kann, ob ein Objekt vor oder hinter ihm ist [25]. Dies ist möglich, da der Abstand zu beiden Ohren gleich ist. Analoges Problem ist, wenn nicht unterschieden werden kann, ob sich das Objekt über oder unter dem Spieler befindet.

In der Anwendung soll die Kamera in der Mitte des 3D-Raumes positioniert werden. Dies kann genutzt werden, um die Geister um den Spieler herum auftauchen zu lassen. Wie in Abbildung 4.1 dargestellt, kann sich der Spieler durch Körperdrehungen im Raum umschauchen.



Abbildung 4.1: Spielprinzip

4.2 Mockups

Um Design-Ideen auszutesten, soll nach Preece et al. die Spielidee aufgezeichnet werden. In frühen Projektphasen eignen sich vor allem Paper-Mockups, da diese schnell und günstig erstellt werden können [17]. Nach Ausarbeitung der Spielidee wurden deshalb Papierprototypen zur Visualisierung der Oberflächen erstellt. Der Einsatz von Mockups an dieser Stelle ist sinnvoll, um einen schnellen Einblick in die zu entwickelnde Anwendung, ohne Programmieraufwand, zu erhalten.

In Abbildung 4.2 ist der Startbildschirm dargestellt. Dieser wird dem Spieler beim Starten der Anwendung angezeigt.

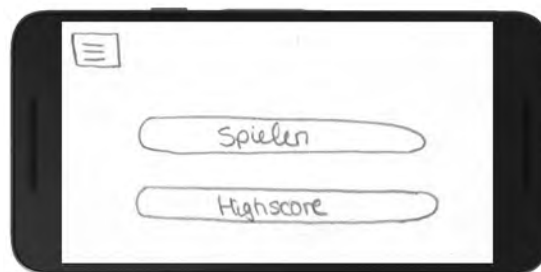


Abbildung 4.2: Startbildschirm

Klickt der Spieler auf den Button 'Spielen' so startet das Spiel (siehe Abbildung 4.3).



Abbildung 4.3: Beispiel einer Spielszene

Während des Spielens sammelt der Spieler Score-Punkte. Diese sollen gespeichert und einsehbar sein. Klickt der Spieler auf den Button 'Highscore' auf dem Startbildschirm, so kann er sich die Highscores ansehen, wie in Abbildung 4.4 dargestellt.

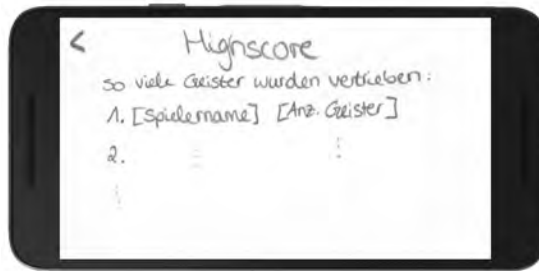


Abbildung 4.4: Highscore

Um das Seitenmenü auszuklappen, muss der Spieler in der linken oberen Ecke auf den Button klicken. In diesem kann der Spieler zusätzliche Funktionen aufrufen und Informationen über die App erhalten.

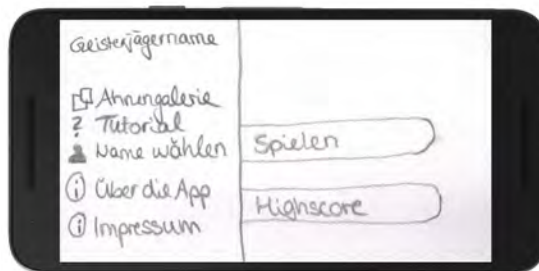


Abbildung 4.5: Erweitertes Menü

Über dieses Menü kommt der Spieler beispielsweise auf die Ahnengalerie, in der die Geister vorgestellt werden sollen (siehe Abbildung 4.6).



Abbildung 4.6: Ahnengalerie

Zusätzlich wurden an dieser Stelle ein paar Geister, die in Abbildung 4.7 zu sehen sind, gezeichnet, um später im Spiel verschiedene Charaktere zu entwickeln.



Abbildung 4.7: Geister

4.3 Navigationskonzept

Um die Abläufe der Bedienung zu erläutern, werden diese nachfolgend in BPMN-Diagrammen [14] dargestellt. Es wurde darauf geachtet, dass alle Bereiche der Anwendung mit wenigen Klicks erreichbar sind.

Wenn der Spieler die Anwendung startet, wird diesem zunächst der Startbildschirm angezeigt. Hier kann er das Spiel starten und die Geister vertreiben. Sobald der Spieler verloren hat, wird der erreichte Score vom System gespeichert. Dieser Ablauf wird in Abbildung 4.8 dargestellt.

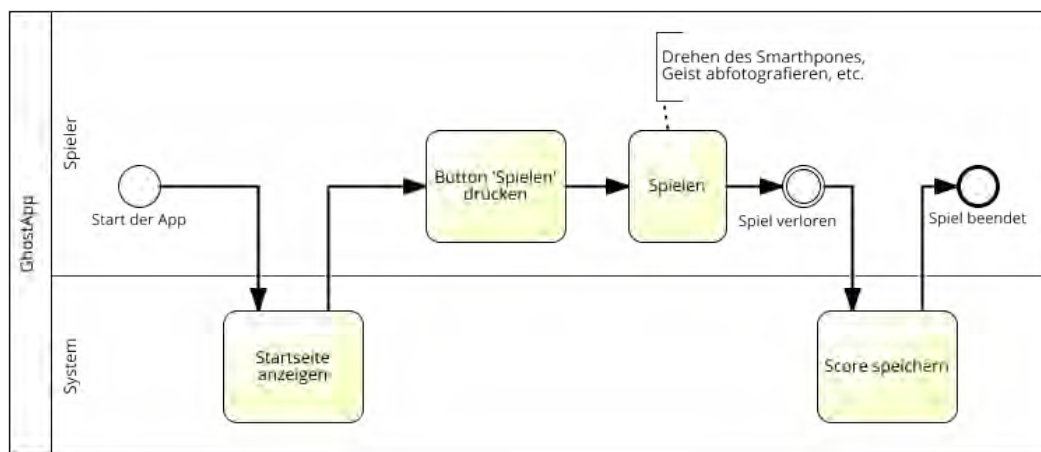


Abbildung 4.8: Start des Spiels

Möchte sich der Spieler diese Bestenliste ansehen, so kann er dies, wie in Abbildung 4.9 gezeigt wird, nach Starten der Anwendung durch Auswahl des Buttons 'Highscore' erreichen. Das System ermittelt die besten fünf Scores und zeigt diese in einer Liste an.

4 Konzeption

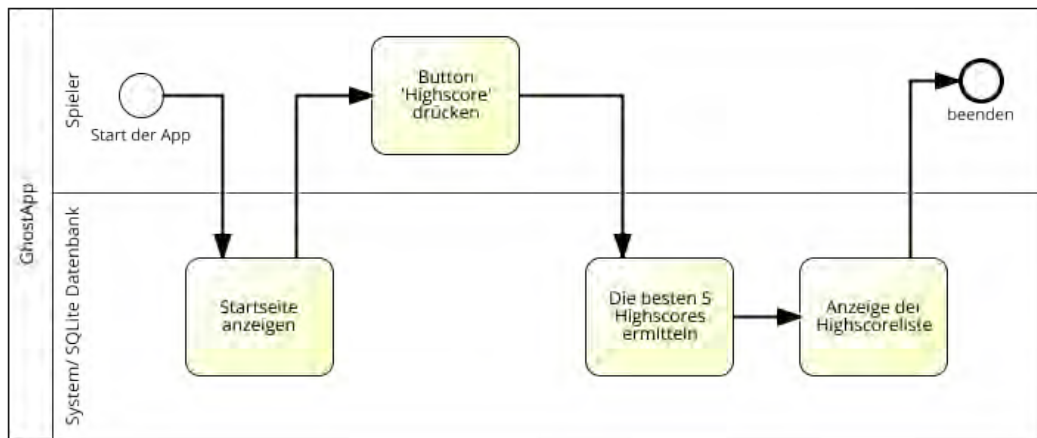


Abbildung 4.9: Bestenliste ansehen

In Abbildung 4.10 ist dargestellt, wie der Spieler seinen gewählten Spielernamen nachträglich ändern kann. Hierfür muss nach Starten der Anwendung das Seitenmenü ausgeklappt und der entsprechende Menüeintrag gewählt werden. Das System zeigt dem Spieler einen neuen Dialog an, in dem der Spieler einen neuen Namen wählt. Sobald dieser bestätigt wird, speichert das System den Namen und zeigt den Startbildschirm mit einem kleinen Begrüßungstext an.

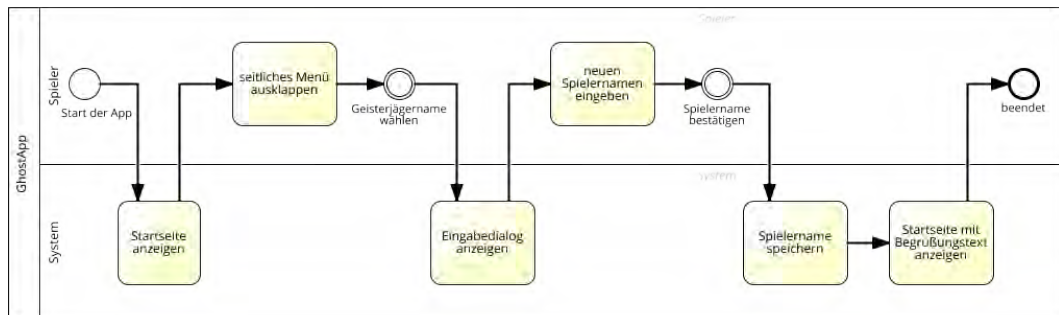


Abbildung 4.10: Spielernamen ändern

4.4 Werkzeuge

4.4.1 Unity

Unity ist eine Engine, mit der 2D- und 3D-Spiele entwickelt werden können. Es handelt sich hierbei um die weltweit führende Echtzeit-Engine, mit der die Hälfte der Spiele weltweit erstellt werden [32]. Als Programmiersprache wird unter anderem C# unterstützt, welche im Rahmen dieser Arbeit verwendet wurde.

Da Unity die technischen Herausforderungen an die zu entwickelnde Anwendung erfüllt, wurde dieses Werkzeug eingesetzt. Ein weiterer wichtiger Grund ist der umfangreiche Asset Store, mit dessen Hilfe die Entwicklung vereinfacht wird [32]. So wurde ein Schloss-Asset aus dem Asset Store integriert. Dieses ermöglicht es, die gewünschten Vorstellungen an die Spielwelt durch Anpassungen umzusetzen. Die Spielkomponente der Anwendung wurde in Unity entwickelt. Dabei war von Vorteil, dass Unity in Szenen arbeitet, in die die Spielelemente eingefügt werden konnten. Die Integration von Modellen, die in Blender [3] erstellt wurden, wird ebenfalls unterstützt. Spielelemente können als sogenannte Prefabs gespeichert werden. Mit Hilfe von Prefabs können so Spielelemente gemeinsam mit ihren Eigenschaften bereitgestellt werden. Für die Entwicklung wurde die Unity-Version 2018.1.6f1 verwendet.

4.4.2 Android Studio

Android Studio ist die offizielle Entwicklungsumgebung zur Entwicklung von Android-Anwendungen. Es ist Gradle-basiert und verfügt über schnelle Emulatoren von handelsüblichen Smartphones mit vielen Funktionen. So können die entwickelten Funktionalitäten direkt am PC getestet werden, ohne das Projekt exportieren und auf das Smartphone übertragen zu müssen [7]. In Android Studio wird in der Programmiersprache Java entwickelt.

Die Programmierung von mobilen Anwendungen erfordert, im Unterschied zu Desktopanwendungen, eine Benutzerinteraktion, die oft nicht eindeutig ist. Dies wird bei der Entwicklung von Android-Anwendungen durch Activity-Klassen erleichtert. Eine Activity stellt ein Fenster bereit, das auf dem mobilen Endgerät als Benutzeroberfläche gezeigt wird.

In dieser Arbeit wurde das Menü einschließlich dessen Funktionalitäten mit der Android Studio Version 3.1.2 erstellt. Über diese Anwendung wird das in Unity erstellte Spiel gestartet.

4.4.3 Blender

Blender ist ein kostenloses, Open Source Werkzeug, mit dem 3D-Modelle, Animationen, Simulationen und vieles mehr erstellt werden können [3]. Von erfahrenen Nutzern können Tools zur Weiterentwicklung von Blender geschrieben werden, so dass es sich bei Blender um ein Community-basiertes Produkt handelt [3].

Neben der Verwendung des 3D-Geisterschlosses ist vor allem die Erstellung der Geister von zentraler Bedeutung. Alle Geister-Modelle wurden in Blender in der Version v2.79 entwickelt. Vorteilhaft ist, dass die Modelle direkt in das Unity-Projekt als Assets eingebunden werden können.

5 Realisierung und Validierung

5.1 Architektur

Im Folgenden wird die Architektur der realisierten Anwendung vorgestellt. Für die Realisierung wurde das in Unity entwickelte Spiel als Library-Modul in das Android Studio Projekt integriert. Als Datenbank wird SQLite [9] verwendet. So ergeben sich in der Anwendung drei entwickelte Komponenten, die in Abbildung 5.1 dargestellt werden.

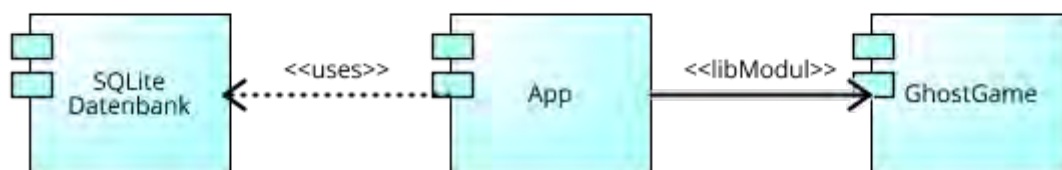


Abbildung 5.1: Komponenten der Anwendung

Über die SQLite Datenbank werden die erreichten Score-Punkte mit dem entsprechenden Spielernamen in der Tabelle 'Highscore' gespeichert. Die zugrundeliegende Relation sieht wie folgt aus:

Highscore(ID, Spielername, Score)

Die Datenbank ist lokal und benötigt somit keine bestehende Internetverbindung. Das Klassendiagramm in Abbildung 5.2 zeigt die beteiligten Klassen. Die Main-Activity-Klasse startet die Klasse 'HighscoreActivity', in der die Bestenliste angezeigt wird. Hierfür werden die Einträge mit Hilfe der Klasse 'HighscoreDBSource', die für alle Datenbankzugriffe zuständig ist, geholt. Um Zugriff auf die Datenbank zu haben, muss diese erst mittels der Klasse 'HighscoreOpenHelper' erzeugt werden. Die Klasse erbt von der SQLiteOpenHelper-Klasse, die die Verbindung zu der SQLite Datenbank herstellt.

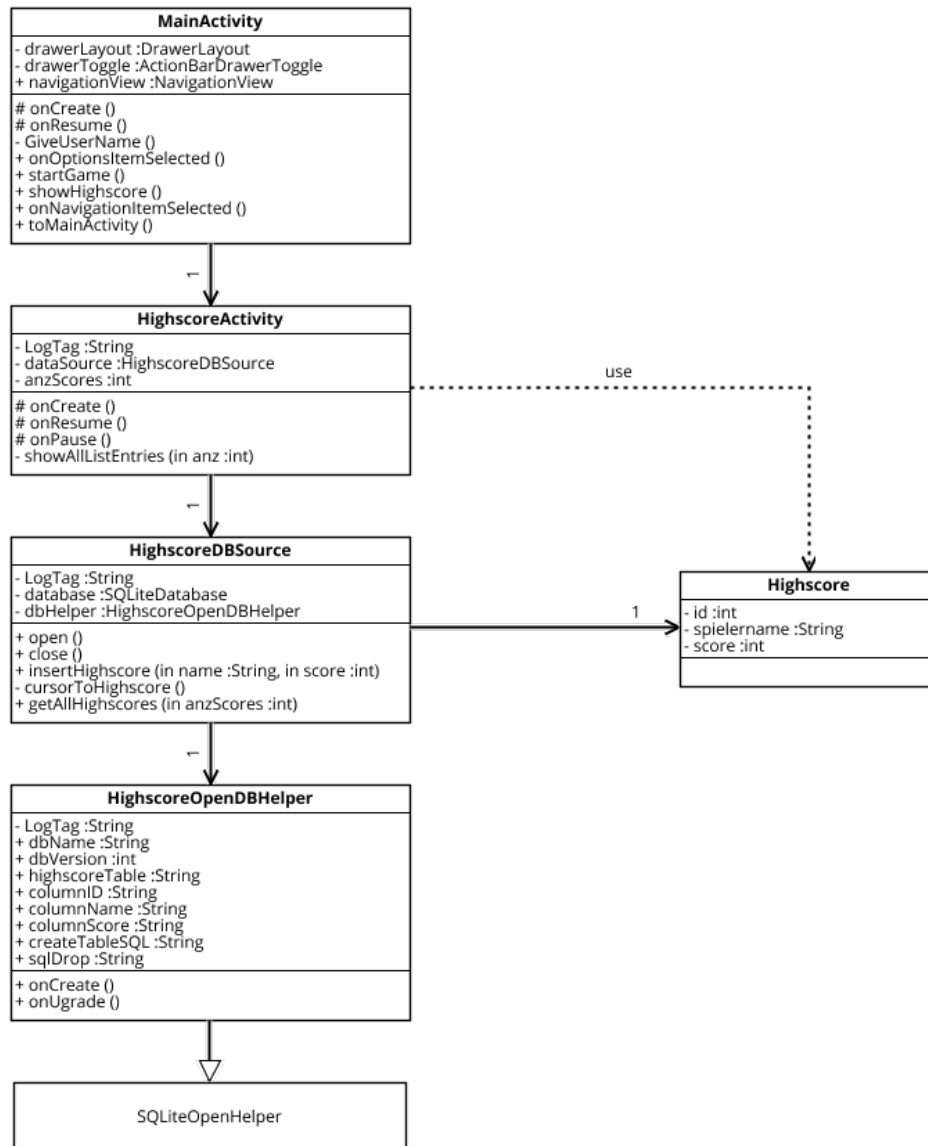


Abbildung 5.2: Klassendiagramm Highscore

Die beteiligten Klassen werden um weitere Klassen des App-Moduls ergänzt und in Abbildung 5.3 dargestellt. Aus Übersichtsgründen werden nur die Namen der Klassen mit den zugehörigen Beziehungen abgebildet. Die zentrale Klasse ist die 'MainActivity'. Diese ist für die Interaktion mit allen weiteren Klassen zuständig. Darunter fällt unter anderem das Aufrufen der Bestenliste und die Auswertung der Benutzerinteraktionen mit dem seitlichen Menü. In der 'MainActivity' wird außerdem

überprüft, ob die Anwendung zum ersten Mal gestartet wurde. Des weiteren ist die Klasse für das Anzeigen des Spielernamens zuständig. Eine zentrale Funktion der Klasse ist das Starten des Library-Moduls über die Klasse 'UnityPlayerActivity'. Die Klassen 'LoginActivity' und 'FirstTimeStartedActivity' sorgen für die Speicherung des Spielernamens. Mit Hilfe der GhostListAdapter-Klasse wird die Anzeige der Ahnengalerie bestimmt. Alle Geister, die in dem Schloss zu finden sind, werden hier angezeigt. Die weiteren Klassen sind zur Darstellung von Spielinformationen zuständig.

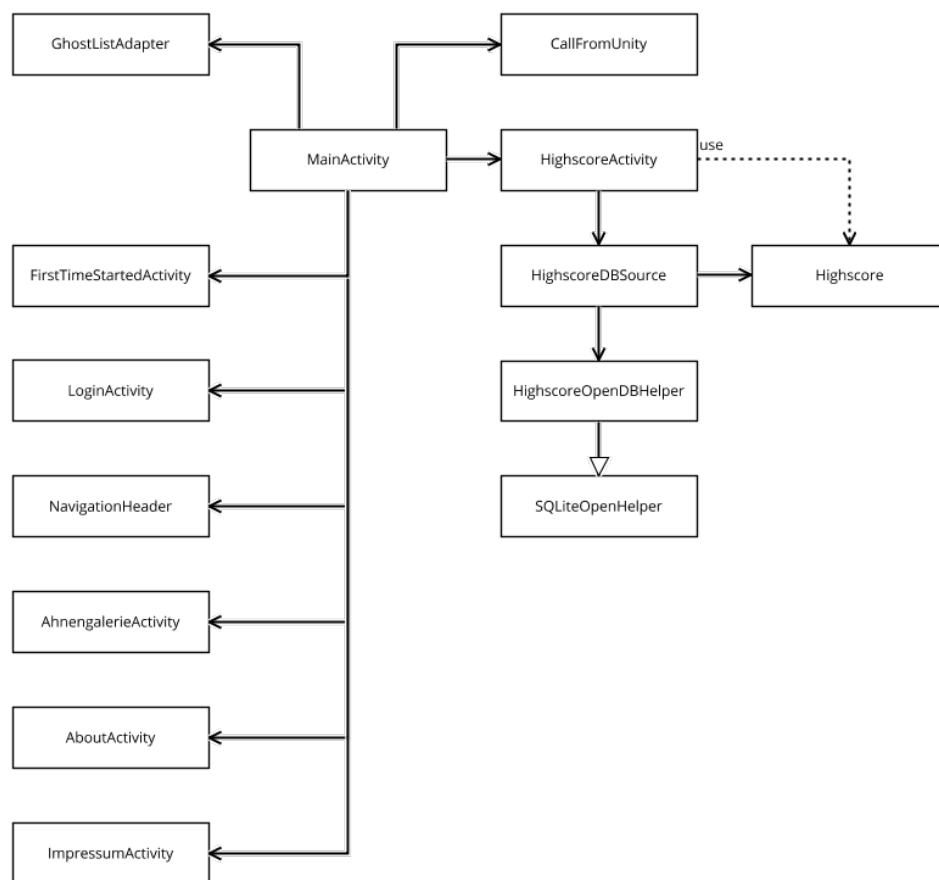


Abbildung 5.3: Klassendiagramm App-Modul

Der Aufbau des Unity-Moduls 'GhostGame' wird in Abbildung 5.4 gezeigt. Das Spiel besteht aus fünf Szenen. Jede Szene enthält eine Kamera, ein Aussehen des Raumes, einen Geist, ein EventSystem und ein Canvas. Die Kamera legt die Raumperspektive fest. Um das Drehen im Raum zu ermöglichen, wird das Skript 'GyroControl' verwendet. Als Spielwelt wird ein Schloss, das auf 'The Big Castle

Kit' [31] aufbaut, verwendet. Insgesamt existieren vier Räume, die aus Elementen des Schloss-Assets bestehen. In den Räumen können die Geister, die als Prefabs gespeichert wurden, auftauchen. Die Positionierung der Geister erfolgt über das Skript 'RandomSpawn'. Den Geistern wird eine Audiodatei zugewiesen, sodass diese hörbar sind. Das EventSystem enthält Skripte, die die Spielmechanik realisieren. Unter anderem wird hier das zufällige Auftauchen der Geister, sowie die Punkteberechnung gesteuert. Mit Hilfe des Canvas wird die Benutzerschnittstelle erstellt. Die Skripte 'PhotoTrigger' und 'GhostPoints' werden benötigt, um den Geist zu fotografieren und im Anschluss die Berechnungen für die Punkte zu realisieren. Über den 'GameManager' wird zur nächsten Szene gewechselt.

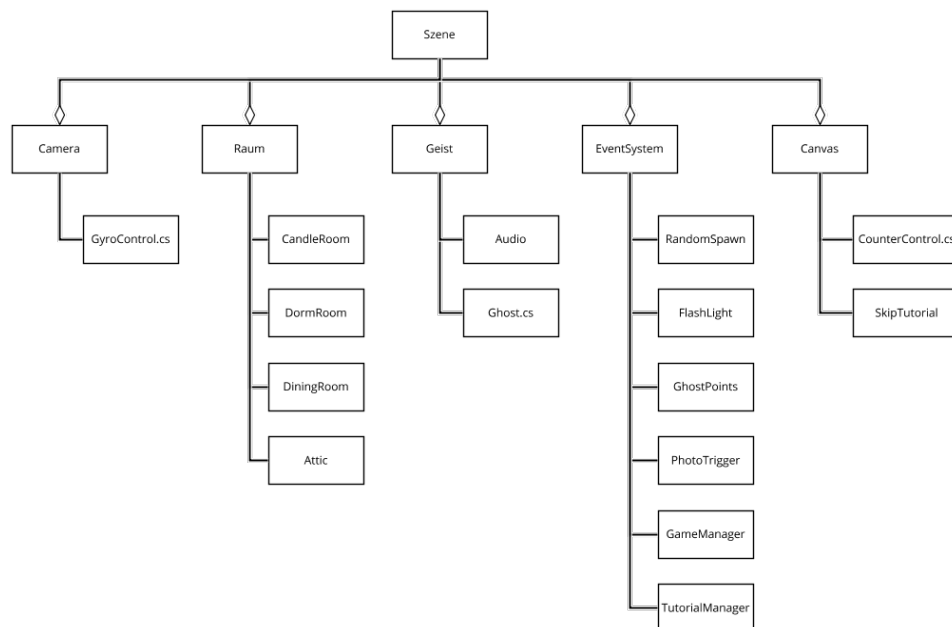


Abbildung 5.4: Aufbau des GhostGames in Unity

Die zugrundeliegende Architektur ermöglicht den Einsatz der Anwendung für ein Smartphone mit Android™ OS-Version 7.1.1 und höher. Dies entspricht der API Version 25 und höher. Weitere Systemvoraussetzungen sind ein Touchdisplay und ein Gyroskop-Sensor sowie angeschlossene Kopfhörer. Dieser muss im Smartphone eingebaut sein, um die Drehungen zu erkennen.

5.2 Ausgewählte Implementierungsdetails

5.2.1 Erstnutzung der Anwendung

Beim ersten Start der Anwendung wird dem Spieler ein Willkommensgruß angezeigt. Dieser beschreibt dem Spieler das Ziel des Spiels und soll einmalig beim ersten Start der Anwendung angezeigt werden. Um herauszufinden, ob das Spiel zum ersten Mal gestartet wird, wird die SharedPreferences API [10] genutzt. Mit dieser wird das Speichern und Laden von Schlüsselwertpaaren für eine bestimmte Anwendung ermöglicht. Die Schlüsselwertpaare werden in einer Datei auf dem Smartphone abgelegt.

Beim Start der Anwendung wird die 'MainActivity' aufgerufen. Es wird ein Schlüsselwertpaar erzeugt, das einen Boolean enthält, ob die 'MainActivity' bereits auf dem Smartphone gestartet wurde. Beim ersten Start wurde sie noch nicht aufgerufen. Deshalb wird eine Activity mit dem Willkommensgruß gestartet und der Schlüsselwert überschrieben. So wird beim nächsten Start sichergestellt, dass die Willkommens-Activity nicht mehr aufgerufen wird, sondern direkt die 'MainActivity'. Wie der Mechanismus mit den SharedPreferences funktioniert, wird in Listing 5.1 dargestellt. Um auf die Daten der SharedPreferences zugreifen zu können, muss zunächst mit Hilfe der Methode 'getDefaultSharedPreferences()' der aktuelle Anwendungskontext definiert werden. Mittels 'getBoolean()' wird der entsprechende Schlüssel übergeben, um den Wert zu erhalten. Mit 'putBoolean()' kann der Wert zu dem entsprechenden Schlüssel editiert werden.

```
1  SharedPreferences prefs = PreferenceManager.  
    ↳ getDefaultSharedPreferences(getBaseContext());  
2  boolean previouslyStarted = prefs.getBoolean(getString(R.  
    ↳ .string.pref_previously_started), false);  
3  if(!previouslyStarted) {  
4      SharedPreferences.Editor edit = prefs.edit();  
5      edit.putBoolean(getString(R.string.  
    ↳ pref_previously_started), Boolean.TRUE);  
6      edit.commit();  
7      GiveUserName();  
8  }
```

Listing 5.1: SharedPreferences für Erstnutzung der App

5.2.2 Spielernamen wählen

Zusätzlich zum Willkommensgruß kann der Spieler in der Willkommens-Activity seinen Spielernamen wählen. Dieser wird im Seitenmenü angezeigt und kann zu einem späteren Zeitpunkt wieder geändert werden. Die Verwaltung des Spielernamens erfolgt über ein weiteres Schlüsselwertpaar in den SharedPreferences. So kann auch von anderen Activities auf diesen zugegriffen werden.

Wie in Listing 5.2 dargestellt, wird der eingegebene Wert dem Schlüssel 'username' zugeordnet und gespeichert. Nach erfolgreicher Speicherung wird dem Spieler eine Nachricht mit dem gewählten Namen, als Rückmeldung, angezeigt. Es wird die Startseite geladen.

```
1  SharedPreferences pref = getSharedPreferences("MyPref",
    ↪ 0);
2  pref.edit().putString("username", textView.getText().
    ↪ toString()).commit();
3
4  Toast.makeText(this, "Hallo " + textView.getText().
    ↪ toString(), Toast.LENGTH_SHORT).show();
5  Intent intent = new Intent(this, MainActivity.class);
6  startActivity(intent);
```

Listing 5.2: Spielernamen

5.2.3 Kopfhöreranschluss

Da das Spiel ohne Ton nicht spielbar ist, soll verhindert werden, dass der Spieler das Spiel startet, ohne Kopfhörer angeschlossen zu haben. Beim Drücken des Spielen-Buttons wird daher der Kopfhöreranschluss geprüft. Sind keine Kopfhörer angeschlossen, wird dem Spieler eine Benachrichtigung angezeigt, die den Spieler auffordert dies nachzuholen. Erst wenn Kopfhörer angeschlossen sind, kann das Spiel gestartet werden. Die Methode wird in Listing 5.3 dargestellt. Mit Hilfe der Methode 'getSystemService()' erhält man eine Referenz auf den AudioManager. Anschließend kann mittels 'isWiredHeadsetOn()' überprüft werden, ob Kopfhörer angeschlossen sind. Für die Benachrichtigung wurde ein eigener Toast entwickelt. Ein Toast ist ein kleines Popup-Element, das eine Rückmeldung anzeigen kann. Sie füllt nur den benötigten Platz aus und die aktuelle Activity bleibt aktiv.

```
1 AudioManager audioManager = (AudioManager)
    ↳ getSystemService(Context.AUDIO_SERVICE);
2
3 if(!audioManager.isWiredHeadsetOn()) {
4     LayoutInflater inflater = getLayoutInflater();
5
6     View layouttoast = inflater.inflate(R.layout.
    ↳ toastcustom, (ViewGroup)findViewById(R.id.
    ↳ toastcustom));
7     ((TextView) layouttoast.findViewById(R.id.texttoast))
    ↳ .setText("Bitte Kopfhoerer anschliessen");
8
9     Toast toast = new Toast(getBaseContext());
10    toast.setView(layouttoast);
11    toast.setDuration(Toast.LENGTH_LONG);
12    toast.show();
13 }
```

Listing 5.3: Kopfhöreranschluss prüfen

5.2.4 Audio

Zur Erzeugung des Sounds für die Geister wurde das SDK 'Resonance Audio' verwendet [8]. Dieses bietet eine plattformübergreifende Unterstützung von 3D-Spatialisierung und der Modellierung komplexer Klangumgebungen an. Da sich Resonance Audio problemlos in Android-Projekte und Unity integrieren lässt, wurde dieses SDK ausgewählt.

Sounddateien, die mit Resonance Audio verwendet werden, verhalten sich genauso, wie reale Schallwellen mit dem menschlichen Ohr und der Umgebung [6]. Diese Schallwellenwechselwirkung wird verwendet, um den Klang lokalisieren zu können (siehe Abbildung 5.5). Mit Hilfe von Resonance Audio kann dies in virtuellen Welten simuliert werden, sodass bestimmte Geräusche hörbar aus einer bestimmten Richtung zu kommen scheinen [6].

Um diese Schallwelleninteraktionen optimal nutzen zu können, verwendet Resonance Audio eine kopfbezogene Übertragungsfunktion [6].

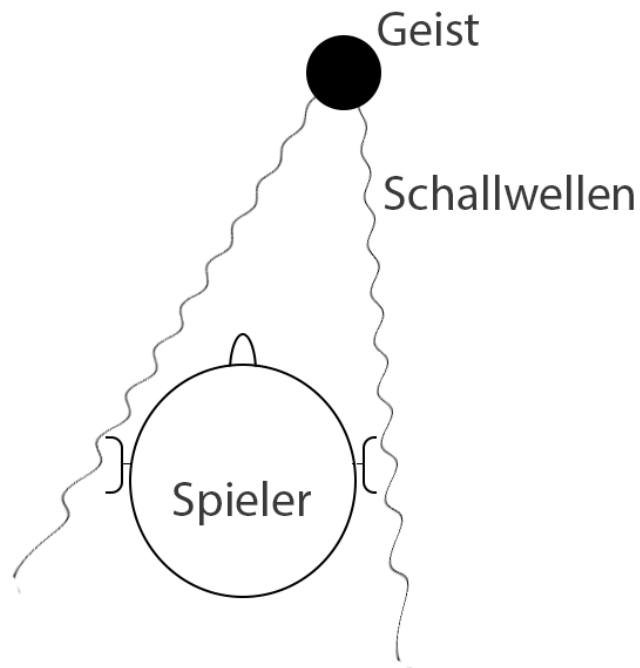


Abbildung 5.5: Wahrnehmung einer Audio-Quelle [13]

5.2.5 Geister

Das Aussehen der Geister wurde in Blender [3] modelliert. Um die Form des Geistes festzulegen, wurde zunächst ein Gerüst aus einer Kugel und zwei Zylindern erstellt (siehe Abbildung 5.6 a). Um das Gewand des Geistes zu erzeugen, wird eine Fläche definiert, die Eigenschaften erhält, sodass sie sich wie ein fließend fallendes Tuch verhält. Dieses Tuch wird oberhalb des Gerüsts platziert und in einer Animation fallen gelassen (siehe Abbildung 5.6 b). So legt sich das Tuch über das Gerüst und bildet die Grundform des Geistes. Das Gerüst kann im Nachgang gelöscht oder ausgeblendet werden. Um dem Geist mehr Leben einzuhauchen, wurden zunächst Augen ausgeschnitten. Im Anschluss daran wurde der Geist mit Hilfe des Sculpt Modes von Blender überarbeitet. Dieser ermöglicht die Veränderung des Modells mit Hilfe von besonderen Pinseln. Diese sind hilfreich, um die Form der Augen zu optimieren und die Form des Geistes anzupassen (siehe Abbildung 5.6 c).

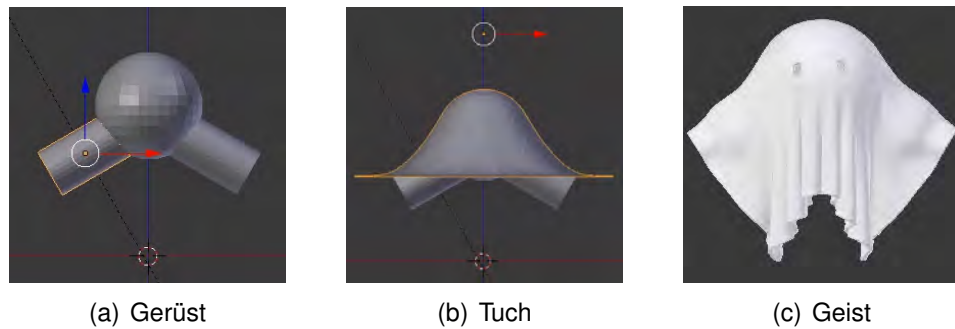


Abbildung 5.6: Entstehung des Geistes

Im Rahmen dieser Arbeit wurden vier verschiedene Geister modelliert. Um jedem Geist ein individuelles Aussehen zu verleihen, wurden diese mit verschiedenen Accessoires ausgestattet. Folgende Elemente wurden hierbei erstellt:

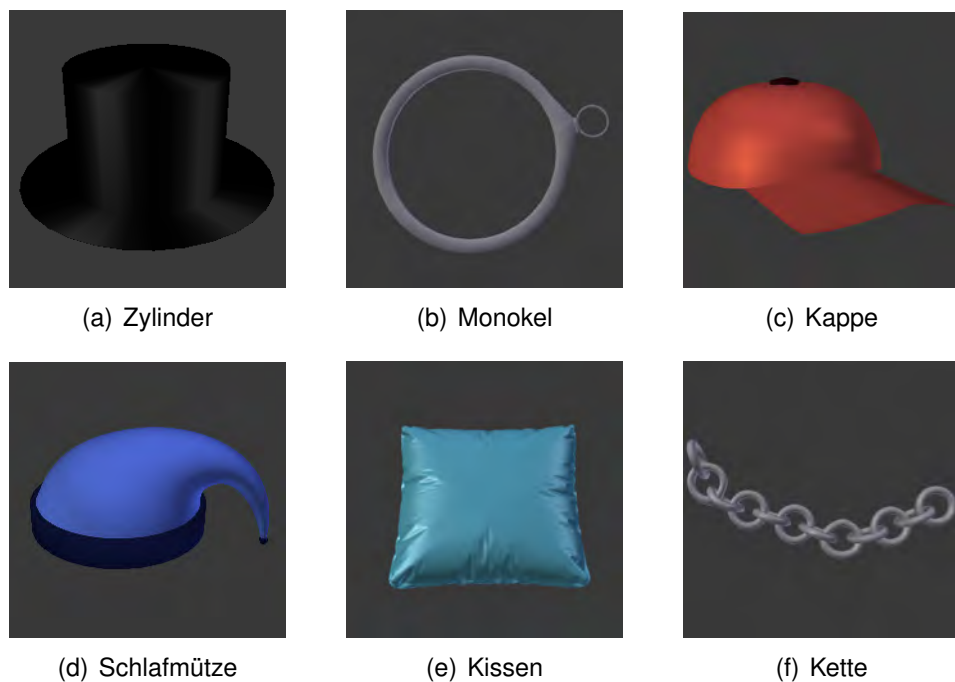


Abbildung 5.7: Accessoires

Die fertigen Geister wurden als fbx-Dateien exportiert und in Unity eingebunden. Dort können diese als Aussehen, in Unity Model genannt, für die Prefabs verwendet werden.

5.2.6 Auftauchen der Geister

In jedem Raum tauchen Geister auf, die vom Spieler gefunden werden müssen. Da der Spieler aus der Sicht der Kamera auf den Raum blickt, wurde diese in der Mitte jedes Raumes positioniert. Damit die Geister rund um den Spieler erzeugt werden, wird ein Kreis um die Kamera gezogen, der die Position der Kamera als Mittelpunkt enthält. Die Position des Geistes wird aus einem zufälligen Winkel und dem Radius berechnet. Die Berechnung wird nachfolgend in Listing 5.4 dargestellt.

```
1 var ang = Random.value * 360;
2 Vector3 pos;
3 pos.x = center.x+radius*Mathf.Sin(ang*Mathf.Deg2Rad);
4 pos.y = center.y;
5 pos.z = center.z+radius*Mathf.Cos(ang*Mathf.Deg2Rad);
```

Listing 5.4: zufällige Position auf Kreis

Diese Position wird genutzt, um das Prefab des jeweiligen Raumes an der Stelle zu erzeugen. Dieses Konzept ist in Abbildung 5.8 dargestellt.

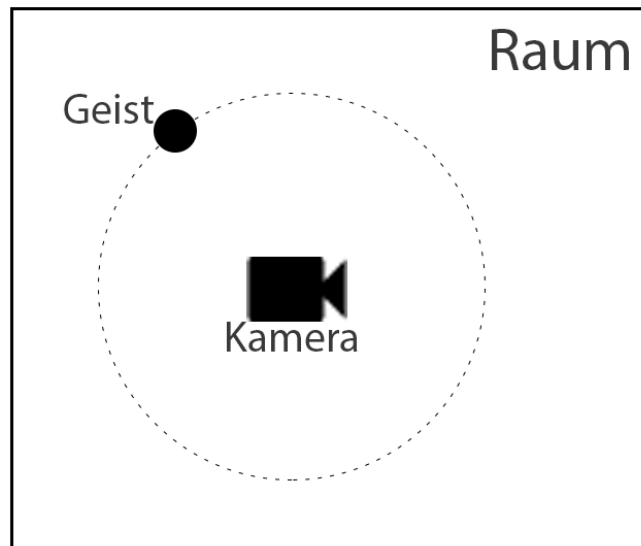


Abbildung 5.8: Konzept: Auftauchen der Geister

Zusätzlich wird der Geist beim Erzeugen in Richtung der Kamera gedreht, so dass der Spieler den Geist von vorne sieht.

5.2.7 Gyroskop

Um die Bewegungen im Raum erkennen zu können, wird das Gyroskop des Smartphones genutzt. Dieser Sensor erkennt Drehungen in horizontaler und vertikaler Richtung. Da das Finden der Geister nur mit Hilfe des Sensors funktioniert, wird beim Drücken des Spielen-Buttons überprüft, ob diese Unterstützung für das verwendete Gerät vorhanden ist. In Listing 5.5 ist die Überprüfung dargestellt. Mit Hilfe des 'PackageManager' und der Methode 'hasSystemFeature()' kann der entsprechende Sensor überprüft werden. Falls die Unterstützung fehlt, wird dem Spieler eine entsprechende Rückmeldung als Toast angezeigt.

```
1 PackageManager packageManager = getPackageManager();
2 boolean gyroExists = packageManager.hasSystemFeature(
   ↪ PackageManager.FEATURE_SENSOR_GYROSCOPE);
3 if(!gyroExists) {
4     View layouttoast=inflater.inflate(R.layout.toast_gyro
   ↪ ,(ViewGroup)findViewById(R.id.toastcustom));
5     ((TextView) layouttoast.findViewById(R.id.texttoast))
   ↪ .setText("Leider keine Gyro-Unterstützung");
6     Toast toast = new Toast(getApplicationContext());
7     toast.setView(layouttoast);
8     toast.show();
9 }
```

Listing 5.5: Gyroskop-Unterstützung prüfen

Ist die Unterstützung des Smartphones gegeben, kann das Spiel gestartet werden. Während des Spielens wird die Kamera basierend auf der Ausrichtung des Smartphones im Raum gedreht. Im nachfolgenden Listing 5.6 wird die Methode dargestellt. 'Update()' wird jedes Frame aufgerufen und ermöglicht eine stetige Anpassung an die Drehungen des Spielers.

```
1 void Update () {
2     if(gyroEnabled) {
3         transform.localRotation = gyro.attitude * rotation;
4     }
5 }
```

Listing 5.6: Drehung

5.2.8 Fotografieren

Meint der Spieler, einen Geist gefunden zu haben, muss er auf eine beliebige Position auf dem Bildschirm tippen. Daraufhin wird das Blitzlicht aktiviert und überprüft, ob sich ein Geist im Blickfeld befindet. Listing 5.7 zeigt die Update-Methode. Wird auf den Bildschirm getippt, wird über die Methode 'ToggleFlashLight()' das Blitzlicht aktiviert. Danach wird mit Hilfe der Methode 'GhostInView()' ermittelt, ob sich ein Geist im Blickfeld befindet, um im Anschluss daran die Score-Punkte in Abhängigkeit von der Position des Geistes berechnen zu können.

```
1 void Update () {
2     if (Input.GetMouseButtonDown(0)) {
3         flashLight.ToggleFlashLight();
4         ghostPoints.GhostInView();
5     }
6 }
```

Listing 5.7: Bildschirmberührungen erkennen

Wenn der Spieler den Blitz auslöst und sich kein Geist in dem Blickfeld der Kamera befindet, erhält der Spieler eine entsprechende Rückmeldung und kann weiter suchen. Befindet sich ein Geist im Blickfeld, so wird die Position des Geistes ermittelt, um die Punkte für den Score zu berechnen. Wenn sich der Geist genau in der Mitte des Sichtfeldes befindet, wurde er optimal getroffen. In diesem Fall erhält der Spieler 10 Punkte. Je größer die Abweichung zu dem Mittelpunkt ist, desto weniger Punkte werden vergeben. In Listing 5.8 ist der Code zur Berechnung der Score-Punkte dargestellt. Zunächst wird aus den Positionen des Geistes und der Kamera die Differenz auf horizontaler Ebene gebildet. Mit Hilfe der Abweichung werden die Punkte berechnet. Diese werden auf den bisherigen Score aufaddiert.

```
1 var ghostAngle=new Vector3(position.x, 0.0f,position.z);
2 var cameraAngle=new Vector3(Camera.main.transform.
   ↪ forward.x,0.0f,Camera.main.transform.forward.z);
3 var horDiffAngle=Vector3.Angle(ghostAngle, cameraAngle);
4 double abweichung=Math.Round(horDiffAngle, 0);
5 double points=Math.Round((((fieldOfView - abweichung)/
   ↪ fieldOfView)*10), 0);
```

Listing 5.8: Berechnung der Punkte

5.2.9 Highscore

Die erreichten Punkte aus dem Spiel werden in einer SQLite Datenbank [9] gespeichert. In Listing 5.9 ist zu sehen, wie dies in dem Unity-Projekt ermöglicht wird. Wenn der Spieler verliert, wird eine Instanz der 'AndroidJavaClass' erzeugt. Diese ruft die Methode 'unityBack()' aus dem Android Studio Modul auf und übergibt als Parameter die Score-Punkte. Über diesen Mechanismus kann zwischen den zwei Modulen, die in unterschiedlichen Programmiersprachen geschrieben wurden, kommuniziert werden.

```
1 AndroidJavaClass javaClass = new AndroidJavaClass("com.  
    ↪ uniulmcompany.jana.ghostapp.CallFromUnity");  
2 javaClass.CallStatic("unityBack", new object[] { scoreUI  
    ↪ .text });
```

Listing 5.9: Aufruf Android-Klasse

In der Methode 'unityBack()' wird als Parameter der erreichte Score übergeben. Dieser wird an die Methode 'insertNewScore()' weitergegeben, um diesen in die Datenbank zu speichern. Dazu wird zunächst eine Verbindung zur Datenbank geöffnet. Dann kann der Score mit dem aktuellen Spielernamen in die Datenbank eingefügt werden. Zum Schluss wird die Verbindung wieder geschlossen und die Startseite angezeigt.

```
1 public static void unityBack(String newScore) {  
2     Integer.getInteger(newScore);  
3     CallFromUnity cFU = new CallFromUnity();  
4     cFU.insertNewScore(newScore);  
5 }  
6 public void insertNewScore(String newScore) {  
7     Toast.makeText(context, "neuer Score: " + newScore,  
8         ↪ Toast.LENGTH_SHORT).show();  
9     int score = Integer.parseInt(newScore);  
10    dataSource.open();  
11    Highscore h = dataSource.insertHighscore(name, score);  
12    dataSource.close();  
13    ((MainActivity)context).toMainActivity();  
14 }
```

Listing 5.10: Score speichern

5.2.10 Ahnengalerie

In der Ahnengalerie werden alle Geister aus dem Geisterschloss dargestellt. So hat der Spieler einen Überblick über alle Geister. Zur Darstellung wird eine dynamische Liste verwendet, dessen Einträge ein Bild, einen Titel und eine Beschreibung enthalten. In Listing 5.11 ist zu sehen, wie die dynamische Liste erzeugt und in der ListView dargestellt wird. Die Parameter der GhostListAdapter-Klasse wurden vorher in Arrays definiert. Der GhostListAdapter erbt von dem ArrayAdapter.

```
1 GhostListAdapter adapter = new GhostListAdapter(this,
    ↪ itemnames, imageids, descriptions);
2 list=(ListView)findViewById(R.id.ghostlist);
3 list.setAdapter(adapter);
```

Listing 5.11: Aufbau der Ahnengalerie

Der Vorteil der dynamischen Liste ist, dass diese leicht mit weiteren Geistern erweitert werden kann. Hierzu müssen ausschließlich die Bilder eingebunden werden und der Geist braucht einen Namen und eine Beschreibung.

5.2.11 Zusammenführung Unity- und Android Studio Projekt

Um das fertige Spiel, das in Unity implementiert wurde, in das in Android Studio entwickelte Menü einzubetten, muss das Unity-Projekt exportiert werden. Die Integration stellt eine besondere Herausforderung dar. Deshalb wird das Vorgehen im Folgenden genau beschrieben.

Schritt 1: Entwicklung Das Unity-Projekt ist angelegt und entwickelt.

Schritt 2: Export In den Build-Settings von Unity muss sichergestellt werden, dass als Plattform Android gewählt ist. Als Build-System muss Gradle ausgewählt werden. Außerdem muss in den Player Settings das Paket und der Name festgelegt werden. Es entsteht ein Gradle-Projekt, das in Android Studio geöffnet werden kann.

Schritt 3: Library Das exportierte Projekt muss in Android Studio geöffnet werden. Wenn Android Studio Gradle-Einstellungen konfigurieren oder Updates vornehmen möchte, sollte diesen zugestimmt werden.

Als erstes muss die gradle-Datei geöffnet werden, um die Anwendung zu einer

Library umzufunktionieren. Hierzu muss die Plugin-Angabe 'application' durch 'library' ersetzt werden:

apply plugin: 'com.android.application' wird zu: *apply plugin: 'com.android.library'*

Außerdem muss die ApplicationId und in der Manifest-Datei der folgende Intent-Code herausgenommen werden.

```
<!--<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
<category android:name="android.intent.category.LEANBACK_LAUNCHER"/>
</intent-filter-->
```

Schritt 4: Import Nun kann die Library als Modul in das Android Studio Projekt eingebunden werden, indem das Projekt in die Projektstruktur hinzugefügt wird. In der Datei 'settings.gradle' muss das Modul aufgenommen werden:

```
include ':app'
include 'GhostGame' //Modulname
```

In der gradle-Datei des App-Moduls muss bei den Abhängigkeiten das Projekt hinzugefügt werden.

```
dependencies {
...
implementation project(':GhostGame')
}
```

In der Manifest-Datei des App-Moduls müssen folgende zwei Zeilen in dem Manifest-Tag und dem Application-Tag angepasst werden:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
package="com.uniulmcompany.jana.ghostapp">
```

```
<application
android:allowBackup="true"
tools:replace="android:icon,android:theme,android:label"
android:icon="@mipmap/ic_launcher"
android:label="@string/title"
...
```

```
</application>  
</manifest>
```

Schritt 5: Activity starten Nun kann das Modul verwendet werden. Die Activity, die das Spiel startet, kann wie folgt aufgerufen werden.

```
Intent intent = new Intent(this, UnityPlayerActivity.class);  
startActivity(intent);
```

5.3 Vorstellung der Anwendung

Nach den technischen Hintergründen soll nun in diesem Abschnitt die entwickelte Anwendung mit Screenshots vorgestellt werden.

Beim ersten Start der Anwendung wird, wie in Abbildung 5.9 zu sehen, ein Willkommensgruß angezeigt. Zusätzlich kann der Spieler einen Spielernamen wählen und diesen durch den Button bestätigen.

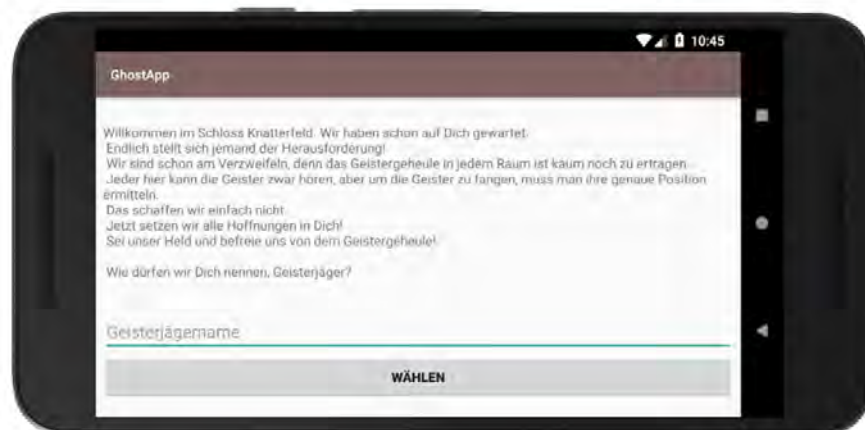
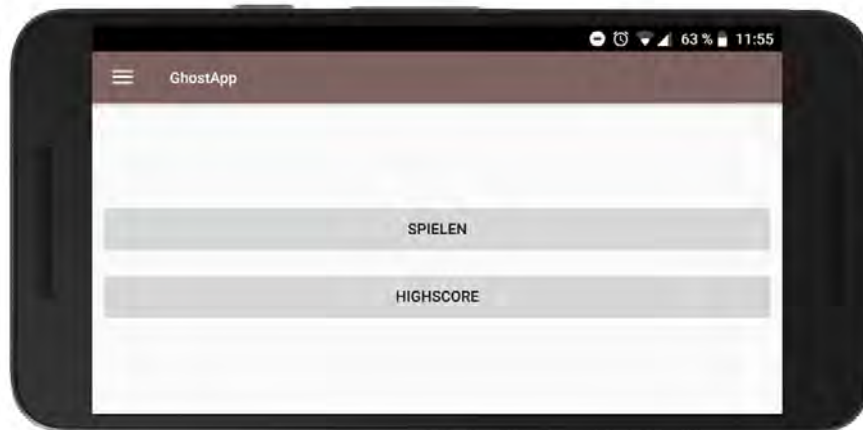
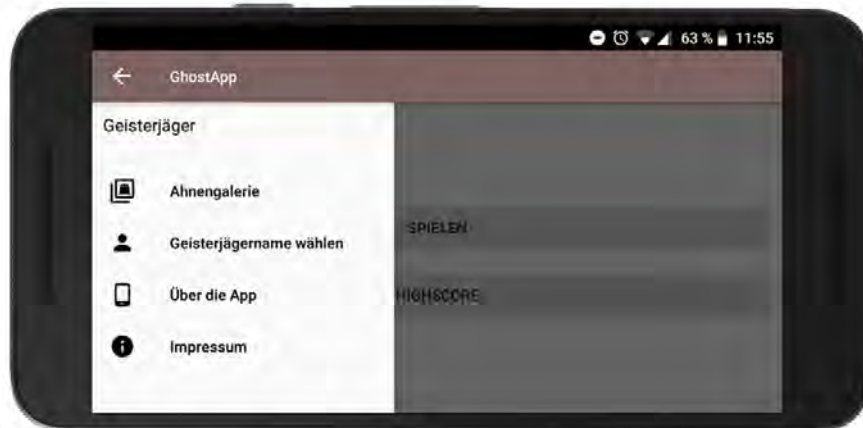


Abbildung 5.9: Erster Start

Nachdem der Spieler seinen Spielernamen gewählt hat, oder die Anwendung ein weiteres Mal gestartet wird, wird dem Spieler die Startseite angezeigt. Von hier aus kann er die komplette Anwendung bedienen. Die zwei Buttons ermöglichen ein schnelles Starten des Spiels und die Anzeige des Highscores. In einem seitlich ausklappbaren Menü kann der Spieler die weiteren Funktionen einsehen. Dies wird in Abbildung 5.10 dargestellt.



(a) Startseite



(b) Seitliches Menü

Abbildung 5.10: Startbildschirm

Drückt der Spieler auf den Spielen-Button, wird das Spiel gestartet und der erste Raum des Geisterschlosses wird angezeigt. Zusätzlich sieht der Spieler die aktuell erreichten Punkte. Wie das Spiel im Spielmodus aussieht, wird in Abbildung 5.11 gezeigt. Um dem Spieler die Spielmechanik näher zu bringen, wird im ersten Raum ein Tutorial vorangestellt. Hier befindet sich der Spieler im ersten Raum des Schlosses, bekommt aber zusätzliche Hilfe angezeigt und kann die Geister sehen. Über einen Button kann das Tutorial jederzeit übersprungen werden.



Abbildung 5.11: Spielmodus

Insgesamt wurden vier Räume entwickelt, die in Abbildung 5.12 dargestellt werden.



(a) Kerzenraum



(b) Schlafsaal



(c) Speisesaal



(d) Dachboden

Abbildung 5.12: Räume aus dem Schloss

In jedem Raum taucht ein anderer Geist auf. In Abbildung 5.13 werden die verschiedenen Geister vorgestellt.



(a) Graf Edwin von Knatterfeld (b) Baron Haribald von Schlafmütze (c) Prinz Leopold I (d) Sir Friedrich von Schreckenstein

Abbildung 5.13: Geister

Von der Startseite aus kann der Spieler auch direkt die Bestenliste, wie in Abbildung 5.14 dargestellt, einsehen. Hierbei werden die Top 5 Scores angezeigt. Dies soll den Spieler motivieren, seinen bisherigen Score zu übertreffen.



Abbildung 5.14: Highscore

Klappt der Spieler das Seitenmenü aus, kann er zur Ahnengalerie gelangen. Hier werden die Geister für den Spieler vorgestellt und geben dem Spiel einen liebevollen Charakter. In Abbildung 5.15 ist der Aufbau der Ahnengalerie zu sehen.



Abbildung 5.15: Ahnengalerie

Zusätzlich kann der Spieler seinen Spielernamen ändern. Hierfür muss der Spieler den entsprechenden Menüeintrag auswählen. Im Anschluss daran kann er seinen Namen eingeben, wie in Abbildung 5.16 zu sehen. Dieser kann dann mithilfe des Wählen-Buttons bestätigt werden.

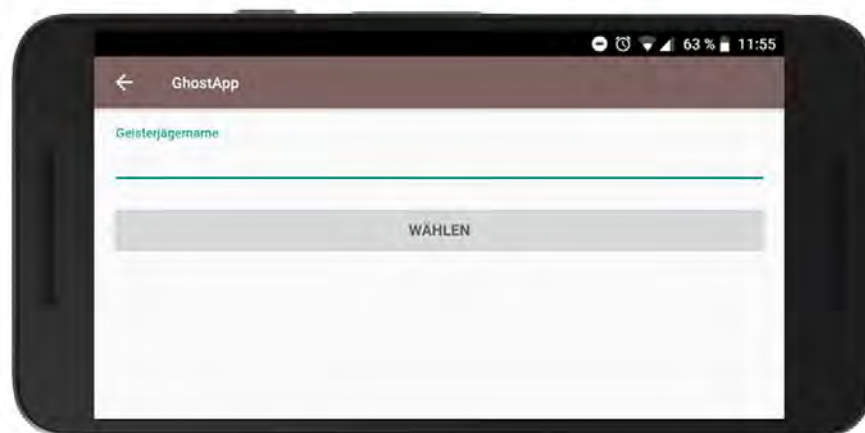
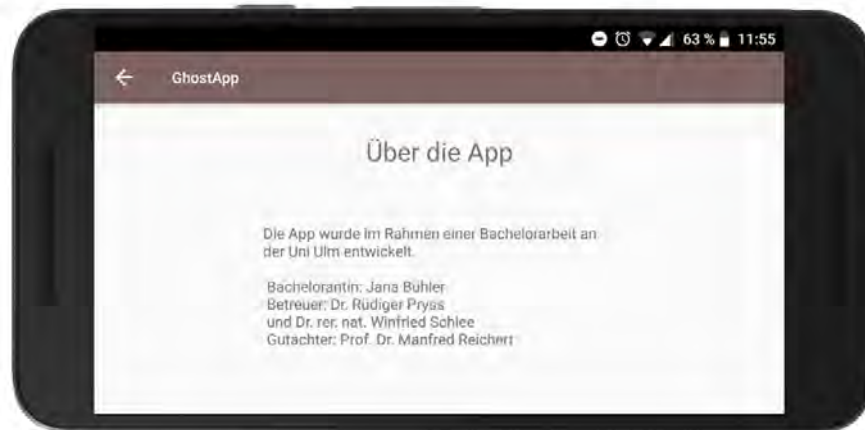


Abbildung 5.16: Spielername ändern

In Abbildung 5.17 sind die Menüeinträge 'About' und 'Impressum' zu sehen. Über diese kann der Spieler zusätzliche Informationen über die Anwendung erfahren.



(a) Über die App



(b) Impressum

Abbildung 5.17: Informationen über die GhostApp

5.4 Erfüllung der Akzeptanzkriterien

Im Folgenden werden in Tabelle 5.1 alle User Stories aufgelistet. Diese wurden kategorisiert und priorisiert. Außerdem wird dargestellt, ob die Akzeptanzkriterien erfüllt wurden.

Wie der Tabelle entnommen werden kann, wurden alle User Stories vollständig umgesetzt.

5 Realisierung und Validierung

	User Story	Beschreibung	Priorität	Akzeptanzkriterium	Erfüllt
Basissystem	US1	App starten	1		✓
				A1	✓
				A2	✓
				A3	✓
	US2	Spielernamen ändern	1	A4	✓
				A1	✓
				A2	✓
				A3	✓
	US4	Spiel spielen	1	A4	✓
				A1	✓
A2				✓	
A3				✓	
A4				✓	
A5				✓	
Features	US3	Highscore	2	A6	✓
				A7	✓
				A8	✓
				A9	✓
	US5	Tutorial	2	A10	✓
				A1	✓
				A2	✓
				A3	✓
				A4	✓
	US6	Ahnengalerie	3	A5	✓
A1				✓	
A2				✓	
US7	Systemvoraussetzungen	3	A1	✓	
			A2	✓	
			A3	✓	

Tabelle 5.1: Erfüllung der Akzeptanzkriterien

6 Fazit

6.1 Zusammenfassung

Ziel dieser Arbeit war die Entwicklung einer Serious Games Anwendung, die es Tinnituspatienten ermöglicht, die Fokussierung von Geräuschen zu trainieren. Außerdem sollte die Anwendung das Orientierungshören verbessern. Wichtig dabei war, dass ein unterhaltsames Spiel entsteht. Es entstand ein mobiles Android-Spiel, das ab der Version 7.1.1 auf Smartphones spielbar ist.

Der Spieler befindet sich in einem Geisterschloss und soll auftauchende Geister fotografieren, um diese verschwinden zu lassen. Die Geister sind unsichtbar. Der Spieler muss durch aufmerksames Hören sein Smartphone in die Richtung drehen, aus der das Geräusch kommt.

Um den Spielcharakter zu stärken, wurde ein dreidimensionales Schloss mit verschiedenen Geistern realisiert. Durch eine Bestenliste konkurriert der Spieler mit sich selbst, um seinen bisherigen Rekord zu übertreffen. Eine Ahnengalerie unterstützt zusätzlich den Spielcharakter. Durch ein kleines Tutorial wird der Spieler mit den Spielmechaniken vertraut gemacht.

Das Spiel wurde hauptsächlich mit Unity und Android Studio entwickelt. Unity wurde für die Spielkomponente benötigt und Android Studio für das Menü sowie die Integration einer SQLite Datenbank. Die beiden Entwicklungsumgebungen wurden für das, wofür sie am besten geeignet sind, verwendet. Dennoch war die Integration des Spiels in das Menü nicht leicht und hat viel Zeit der Arbeit in Anspruch genommen.

Zusammenfassend lässt sich sagen, dass das entstandene Serious Game neben dem Training vor allem Spaß macht. Das Ziel eines Spiels zum Trainieren des Richtungshörens wurde erreicht. Die in den User Stories festgehaltenen Anforderungen wurden erfüllt.

6.2 Ausblick

Im Rahmen dieser Bachelorarbeit entstand ein spielbares Serious Game, in das zukünftige Erweiterungen integriert werden können. Nachfolgend werden diese genauer erläutert.

Besonders interessant für Tinnituspatienten wäre es, wenn der Ton des Geistes in seiner Frequenz individuell anpassbar wäre. Da die Tinnituswahrnehmung subjektiv ist, kann so der Trainingseffekt verbessert werden. Zusätzlich könnte man mit Fragebögen arbeiten, die in Spielpausen angezeigt werden. Hierbei könnten Fragen zu der aktuellen Stärke der Beschwerden und dem Befinden des Spielers gestellt werden. Durch Auswertungen der Fragebögen können neue Erkenntnisse gewonnen werden.

Um den Spielcharakter zu erweitern, könnte man zusätzliche, seltene Geister integrieren, die nicht regelmäßig auftauchen. Sie geben Bonuspunkte oder verschaffen dem Spieler weitere Vorteile beim Spielen. In der Ahnengalerie könnten noch nicht gefundene Geister ausgegraut werden. So hat der Spieler die weitere Motivation, möglichst alle Geister zu finden und trainiert hierdurch mehr.

Da bisher nur der Kopfhöreranschluss des Smartphones überprüft wird, wäre die Erkennung von Bluetooth-Kopfhörern interessant.

Nach Implementierung der Erweiterungen wäre die Durchführung einer Benutzerstudie sinnvoll. Hierdurch wäre es möglich, ein aussagefähiges Feedback zu der GhostApp und deren Nutzung zu erhalten. Nach erfolgreichem Abschluss der Studie kann die Anwendung im Google Play Store oder einem anderen App Store zum Download bereitgestellt werden.

Literaturverzeichnis

- [1] BECK, K. ; AL. et: *Manifest für Agile Softwareentwicklung*. <http://agilemanifesto.org/iso/de/manifesto.html>, 2001. – zuletzt besucht am: 10.09.2018
- [2] BIESINGER, E. ; IRO, H. : *HNO Praxis heute, Tinnitus*. Bd. 25. 2005. – ISBN 3540227202
- [3] BLENDER FOUNDATION: *About*. <https://www.blender.org/about/>, 2018. – zuletzt besucht am: 15.09.18
- [4] DÖRNER, R. ; GÖBEL, S. ; EFFELSBERG, W. ; WIEMEYER, J. : *Serious games: foundations, concepts and practice*. Springer, 2016. – ISBN 9783319406114
- [5] GOLL, J. ; HOMMEL, D. : *Mit Scrum zum gewünschten System*. Springer, 2015. – ISBN 9783658107208
- [6] GOOGLE DEVELOPERS: *Fundamental Concepts*. https://developers.google.com/resonance-audio/discover/concepts#simulating_sound_waves_interacting_with_human_ears, 2018. – zuletzt besucht am: 15.09.18
- [7] GOOGLE DEVELOPERS: *Meet Android Studio*. <https://developer.android.com/studio/intro/>, 2018. – zuletzt besucht am: 15.09.18
- [8] GOOGLE DEVELOPERS: *Overview*. <https://developers.google.com/resonance-audio/develop/overview>, 2018. – zuletzt besucht am: 15.09.18
- [9] GOOGLE DEVELOPERS: *Save data using SQLite*. <https://developer.android.com/training/data-storage/sqlite>, 2018. – zuletzt besucht am: 20.10.18
- [10] GOOGLE DEVELOPERS: *SharedPreferences*. <https://developer.android.com/reference/android/content/SharedPreferences>, 2018. – zuletzt besucht am: 17.10.18

- [11] HAMANN, B. : *Tinnitus natürlich heilen: Erfolgreiche Therapien gegen die quälenden Ohrgeräusche*. Kopp Verlag, 2015. – ISBN 9783864452642
- [12] HESSE, G. ; BIESINGER, E. ; GREIMEL, K. ; LAUBERT, A. ; NELTING, M. ; SCHAAF, H. ; WEDEL, H. von: *Retraining und Tinnitustherapie*. 1999. – ISBN 313123461x
- [13] NUORA, J. u. a.: Introduction to sound design for virtual reality games: a look into 3D sound, spatializer plugins and their implementation in Unity game engine. (2018)
- [14] OMG: *BPMN v2.0*. <http://www.bpmn.org/>, 2011. – zuletzt besucht am: 19.09.18
- [15] PIVOTAL SOFTWARE: *Pivotal Tracker*. <https://www.pivotaltracker.com/>, 2018. – zuletzt besucht am: 12.09.18
- [16] PIVOTAL SOFTWARE: *Pivotal Tracker: Quick start & demos*. https://www.pivotaltracker.com/help/articles/quick_start/, 2018. – zuletzt besucht am: 12.09.18
- [17] PREECE, J. ; ROGERS, I. ; SHARP, H. ; BENYON, D. ; HOLLAND, S. ; CAREY, T. : *HUMAN-COMPUTER INTERACTION*. Addison-Wesley, 1994. – ISBN 0201627698
- [18] PROBST, T. ; PRYSS, R. ; LANGGUTH, B. ; SCHLEE, W. : Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the “TrackYourTinnitus“ application. In: *Scientific Reports* 6 (2016), February. <http://dbis.eprints.uni-ulm.de/1396/>
- [19] PRYSS, R. ; PROBST, T. ; SCHLEE, W. ; SCHOBEL, J. ; LANGGUTH, B. ; NEFF, P. ; SPILIOPOULOU, M. ; REICHERT, M. : Mobile Crowdsensing for the Juxtaposition of Realtime Assessments and Retrospective Reporting for Neuropsychiatric Symptoms. In: *30th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2017)*, IEEE Computer Society Press
- [20] PRYSS, R. ; PROBST, T. ; SCHLEE, W. ; SCHOBEL, J. ; LANGGUTH, B. ; NEFF, P. ; SPILIOPOULOU, M. ; REICHERT, M. : Prospective crowdsensing versus retrospective ratings of tinnitus variability and tinnitus – stress associations based on the TrackYourTinnitus mobile platform. In: *International Journal of Data Science and Analytics* (2018), March. <http://dbis.eprints.uni-ulm.de/1654/>

- [21] PRYSS, R. ; REICHERT, M. ; LANGGUTH, B. ; SCHLEE, W. : Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research. In: *IEEE 4th International Conference on Mobile Services (MS 2015)*, IEEE Computer Society Press, 352–359
- [22] PRYSS, R. ; REICHERT, M. ; SCHLEE, W. ; SPILIOPOULOU, M. ; LANGGUTH, B. ; PROBST, T. : Differences between Android and iOS Users of the TrackYourTinnitus Mobile Crowdsensing mHealth Platform. In: *31th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2018)*, IEEE Computer Society Press, 411–416
- [23] PRYSS, R. ; SCHLEE, W. ; LANGGUTH, B. ; REICHERT, M. : Mobile Crowdsensing Services for Tinnitus Assessment and Patient Feedback. In: *6th IEEE International Conference on AI & Mobile Services (IEEE AIMS 2017)*, IEEE Computer Society Press
- [24] PRYSS, R. ; SCHOBEL, J. ; REICHERT, M. : Requirements for a Flexible and Generic API Enabling Mobile Crowdsensing mHealth Applications. In: *4th Int'l Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS), RE'18 Workshops*
- [25] SCHICKLER, M. ; PRYSS, R. ; REICHERT, M. ; SCHOBEL, J. ; LANGGUTH, B. ; SCHLEE, W. : Using mobile serious games in the context of chronic disorders: a mobile game concept for the treatment of tinnitus. In: *Computer-Based Medical Systems (CBMS), 2016 IEEE 29th International Symposium on IEEE*, 2016, S. 343–348
- [26] SCHLEE, W. ; PRYSS, R. ; PROBST, T. ; SCHOBEL, J. ; BACHMEIER, A. ; REICHERT, M. ; LANGGUTH, B. : Measuring the Moment-to-Moment Variability of Tinnitus: The TrackYourTinnitus Smart Phone App. In: *Frontiers in Aging Neuroscience* 8 (2016), December, 294–294. <http://dbis.eprints.uni-ulm.de/1531/>
- [27] SCRUM.ORG: *Der Scrum Guide TM, Der gültige Leitfaden für Scrum: Die Spielregeln*. <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>, 2017. – zuletzt besucht am: 10.09.2018
- [28] STATISTA: *Statista 2017: Welches Kostüm werden Sie in diesem Jahr zu Halloween tragen?* <https://de.statista.com/statistik/daten/studie/763077/umfrage/umfrage-zu-den-beliebtsten-kostuemen-an-halloween-in-deutschland/>, 2017. – zuletzt besucht am: 24.09.2018

LITERATURVERZEICHNIS

- [29] STATISTA: *Statista 2018: Unter welchen der folgenden Beschwerden leiden Sie mindestens gelegentlich?* <https://de.statista.com/statistik/daten/studie/671474/umfrage/umfrage-zu-gesundheitlichen-beschwerden-in-deutschland/>, 2018. – zuletzt besucht am: 10.09.2018
- [30] TECHNISCHE UNIVERSITÄT DARMSTADT: *Der will doch nur spielen - Serious Games.* https://www.youtube.com/watch?v=NmAbbT_CjM8, 2011. – zuletzt besucht am: 10.09.18
- [31] UNITY: *The Big Castle Kit.* <https://assetstore.unity.com/packages/3d/environments/historic/the-big-castle-kit-75818>, 2018. – zuletzt besucht am: 27.7.18
- [32] UNITY TECHNOLOGIES: *The world's leading real-time engine.* <https://unity3d.com/de/unity>, 2018. – zuletzt besucht am: 15.09.18
- [33] WATTANASOONTORN, V. ; HERNÁNDEZ, R. ; SBERT, M. : *Serious games for e-health care.* Springer, 2014. – ISBN 9789814560313
- [34] WIEMEYER, J. : *Serious Games für die Gesundheit.* Springer, 2016. – ISBN 9783658154714
- [35] WISE, K. ; KOBAYASHIA, K. ; SEARCHFIELD, G. : Feasibility study of a game integrating assessment and therapy of tinnitus. In: *Journal of neuroscience methods* (2015), S. 1–7

Name: Jana Bühler

Matrikelnummer: 871153

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Jana Bühler