



Implementation and evaluation of a mobile Windows-application for auditory stimulation of chronic tinnitus patients

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Martin Weidhaas
martin.weidhaas@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Marc Schickler

2015

Fassung November 16, 2015

© 2015 Martin Weidhaas

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Satz: PDF- \LaTeX 2 ϵ

Abstract

Von chronischen Tinnitus betroffene Menschen müssen jeden Moment mit einem Geräusch leben, welches auf keine physikalische Quelle zurückzuführen ist. Dieses Leiden kann sogar zu Depressionen oder Arbeitsunfähigkeit führen. Die Erforschung von Tinnitus steht noch am Anfang und es existiert kein Heilmittel. Lediglich längerfristige Therapien und Trainings können den Betroffenen helfen zu lernen mit ihren Ohrgeräuschen zu leben. Im Institut für Datenbanken und Informationssysteme der Universität Ulm wurde die mobile Anwendung "Track your Tinnitus" entwickelt mit deren Hilfe die Tinnitusintensität über den Tagesverlauf hinweg vom Betroffenen aufgezeichnet werden kann.

Um diese Anwendung zukünftig um einen Gehörtrainingsaspekt zu erweitern wird in der vorliegenden Arbeit eine mobile Anwendung für Windows Phone zur auditorischen Stimulation von chronischen Tinnituspatienten vom Konzept bis zu einem lauffähigen Prototypen erarbeitet. Dabei muss der Benutzer ein oder zwei Tiergeräusche, welche mittels virtuellem surround sound aus einer zufälligen Richtung wiedergegeben werden, orten. Dazu kann sich der Benutzer drehen, um das Geräusch vor sich zu positionieren um anschließend ein "Foto" des gehörten Tieres zu machen. Die benötigte Zeit, sowie die Abweichung zum Geräusch werden zur Laufzeit in Logfiles dokumentiert.

In einer Studie wurde der Einfluss der gewählten Geräusche auf die Präzision und die benötigte Zeit untersucht. Dabei wurde festgestellt, dass sich die Anzahl der Geräusche nicht signifikant auf die Zeit oder Präzision auswirken. Jedoch wurde eines der verwendeten Geräusche signifikant präziser geortet, als das andere. Somit wurde gezeigt, dass die Geräuschauswahl für zukünftige Implementierungen des Trainingsmoduls der "Track your Tinnitus" Anwendung eine sehr wichtige Rolle spielt.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziel der Arbeit	2
1.2	Aufbau der Arbeit	2
2	Grundlagen	5
2.1	Tinnitus	5
2.2	Räumliche Lokalisierung von Geräuschen	7
2.2.1	Head-related Transfer Funktionen (HRTF)	9
2.2.2	Auftretende Probleme mit HRTFs	11
3	Implementierung	13
3.1	Konzept der Anwendung	13
3.2	Anforderungsanalyse	15
3.2.1	Funktionale Anforderungen	15
3.2.2	Nicht-funktionale Anforderungen	16
3.2.3	Styleguide	16
3.2.4	Weitere Features	17
3.2.5	Bewertung und Einteilung der Features	18
3.2.6	Anforderungen an das verwendete Smartphone	18
3.3	Konzept des User Interface	19
3.3.1	User Interaktionen	21
3.4	Verwendete Hardware und Entwicklertools	22
3.4.1	Hardware	22
3.4.2	Tools	23
3.5	Herausforderungen und Probleme während der Implementierung	23
3.5.1	Aufgetretene Probleme	24
3.5.2	Umsetzung der wichtigsten Kernelemente	24
3.6	Vergleich zwischen Applikation und Konzept	30
3.6.1	Spielelemente	30

3.7	Dokumentation für die Wiederverwendbarkeit	35
3.7.1	Softwarekonfigurationen	35
3.7.2	Die Anwendung auf einem Windows Phone installieren	35
3.7.3	Logfiles	36
3.7.4	Codestruktur	37
3.7.5	Bekannte Probleme	41
4	Studie	43
4.1	Aufbau	43
4.2	Ablauf	44
4.3	Hypothesen	44
4.4	Ergebnisse	44
4.4.1	Bearbeitungszeiten	45
4.4.2	Winkeldifferenzen	45
4.4.3	Vergleich aller Betriebssysteme	46
4.5	Diskussion der Ergebnisse	47
4.5.1	Schwächen der Studie	47
4.5.2	Betrachtung der Hypothesen	48
4.5.3	Vergleich zwischen den Plattformen	49
5	Zusammenfassung	51
5.1	Implementierung	51
5.1.1	Realisierung der Anwendung	51
5.1.2	Anforderungsabgleich	52
5.1.3	Tools	52
6	Fazit und Ausblick	55
6.1	Fazit aus der Studie	55
6.2	Ausblick	55
	Literatur	59
A	Anhang	63
A.1	Fragebogen der begleitenden Studie	64

Abbildungsverzeichnis

2.1	An beiden Ohren ankommende Schallwellen. Obere Reihe: 30 Grad Abweichung zur Front des Hörers. Untere Reihe: 90 Grad Abweichung [32]	8
2.2	Frequenzspektren an beiden Ohren. Oben: 30 Grad Abweichung. Unten: 90 Grad Abweichung [32]	8
2.3	Cone of confusion [21]	9
2.4	Versuchsaufbau zur Messung von HRTFs [32]	10
2.5	Schematische Darstellung der Übertragung eines Geräusches von einer Quelle zu den Ohren [32]	11
3.1	Sichtbarer Ausschnitt einer virtuellen Umgebung	14
3.2	Verwendetes Farbschema	17
3.3	Nokia Lumia 830 (rechts) [18]	23
3.4	Veranschaulichung der Achsen eines Körpers [20]	25
3.5	Veranschaulichung des Koordinatensystems zur Positionierung der Geräusche	27
3.6	Sichtbarer Bildausschnitt aus dem Hintergrundbild	30
3.7	Frequenzanalyse des Vogelrufes	31
3.8	Frequenzanalyse des Froschrufes	31
3.9	Bild eines Blauhähers für das erste Geräusch Quelle: https://flic.kr/p/9Kj8gX	32
3.10	Bild eines Frosches für das zweite Geräusch Quelle: https://flic.kr/p/8mFkuV	32
3.11	Verwendetes Panoramabild für den Hintergrund Quelle: https://commons.wikimedia.org/wiki/File:Goat_Peak,_Cascades.jpg	32
3.12	Piktogramm einer Kamera als Schaltfläche während der Spielphase	32
3.13	Seite zum Eingeben des Studienrelevanten Benutzer-Codes	33
3.14	Hauptmenü Seite	33
3.15	Bildschirmfoto der Spielphase bei der Suche des ersten Geräusches	34
3.16	Ergebnis-Seite bei der Suche des ersten Geräusches	34

Abbildungsverzeichnis

3.17 Ergebnis-Seite bei der Suche nach dem zweiten Geräusch	34
3.18 Menüleiste zum Exportieren der Anwendung in Visual Studio	35
3.19 Schematischer Aufbau des Quellcodes	40
4.1 Zeit- und Winkeldifferenzunterschiede zwischen zu suchenden Geräuschen	47
4.2 Zeit- und Winkeldifferenzunterschiede zwischen den Betriebssystemen .	48

Tabellenverzeichnis

3.1	Muss-, Soll-, Kann-Klassifikation der Features	19
4.1	Varianzanalyse der benötigten Zeit	45
4.2	Gemittelte benötigte Zeit zum suchen der Ziele	45
4.3	Varianzanalyse der absoluten Winkeldifferenz	46
4.4	Gemittelte absolute Winkeldifferenz der Ziele	46
4.5	Bearbeitungszeiten und Winkeldifferenzen der einzelnen Betriebssysteme im Vergleich	47
5.1	Muss-, Soll-, Kann-Klassifikation der Features mit Implementierungsstatus	52

1 Einleitung

Tinnitus ist eine Krankheit, welche die auditorische Wahrnehmung stark beeinträchtigen kann. Sie kann zu Folgeerkrankungen, wie beispielsweise Schlaf- und Konzentrationsstörungen führen und bei betroffenen Personen Depressionen auslösen. Dies kann bis zur Arbeitsunfähigkeit führen.

Deshalb ist es für Betroffene von großer Bedeutung zu lernen, wie ein Leben mit Tinnitus ohne Einschnitte in den Alltag möglich ist. Das Institut für Datenbanken und Informationssysteme der Universität Ulm kooperiert schon seit einiger Zeit mit der Tinnitus Research Initiative (TRI), um die mobile Anwendung "Track your Tinnitus" zu entwickeln [10]. Mit dieser mobilen Anwendung wird Menschen mit Tinnitus eine Möglichkeit gegeben ihre Ohrgeräusche besser zu verstehen und sich besser mit ihnen zu arrangieren. Der Anwender kann direkt mit seinem Smartphone Zeitpunkte in seinem Tagesablauf festhalten, bei denen der Tinnitus besonders lästig oder kaum wahrnehmbar war. Somit kann nachvollzogen werden, zu welchen Zeiten oder zu welchen Gelegenheiten die Tinnituserfahrung besonders störend war, sodass der Betroffene solche Situationen in Zukunft besser vermeiden kann.

Die mobile Anwendung ist momentan für Googles Android und für Apples iOS verfügbar [23]. Jedoch existieren bereits Pläne, die Verfügbarkeit der Anwendung zu verbessern. So soll "Track your Tinnitus" in Zukunft ebenfalls für das Windows Phone verfügbar sein. Ebenso wird über eine online verfügbare Umsetzung mittels HTML5 nachgedacht.

Im Institut für Datenbanken und Informationssysteme der Universität Ulm war schon seit längerem geplant, die "Track your Tinnitus" Anwendung zu erweitern. Neben der bisher alleinigen Aufzeichnung des Tinnitus, soll es durch integrierte Anwendungen möglich werden, das Gehör ebenfalls zu trainieren. So sollen beispielsweise kleine Spiele in die Anwendung integriert werden, welche die Aufmerksamkeit des Spielers auf den auditorischen Input lenken und den Tinnitus somit subjektiv schwächer wahrnehmbar machen.

Als Inspiration für solch eine Erweiterung diente die mobile Anwendung "Audio Defence: Zombie Arena" für iOS. In diesem Spiel muss sich der Spieler gegen herannahende

Zombies zur Wehr setzen. Jedoch kann er währenddessen nichts sehen, sondern muss sich ausschließlich auf sein Gehör verlassen [1]. Um die Richtung zu bestimmen, aus der die Zombies auf ihn zukommen, muss sich der Spieler mit seinem Smartphone drehen, bis er den Zombie vor sich hört. Anschließend kann der Zombie - sofern der Spieler richtig gezielt hat - mit einer Berührung auf dem Bildschirm besiegt werden.

1.1 Ziel der Arbeit

Diese Form der auditorischen Stimulation kann ebenfalls für eine spielerische Behandlung von chronischem Tinnitus verwendet werden [3]. Dies möchte sich die in dieser Arbeit zu konzipierende Anwendung für Windows Phone zu Nutze machen. Dem Benutzer werden dabei über Kopfhörer Geräusche aus einer gewissen Richtung vorgespielt. Nun hat dieser die Aufgabe das Geräusch mit Hilfe seines Smartphones zu lokalisieren, indem er sich in die Richtung, aus der das Geräusch kommt dreht. Diese ungewöhnliche Art der Spielsteuerung wird durch die in mittlerweile fast allen mobilen Geräten verbauten Sensoren ermöglicht und liefert eine neuartige Nutzererfahrung [27].

Da die Anwendung für alle Altersgruppen geeignet sein soll, wird die Erweiterung mit einem gewaltfreien Spielgegenstand realisiert. Ziel dieser Arbeit ist die Realisierung eines solchen Prototyps für Windows Phone. Parallel wird eine ähnliche Anwendung mit dem selben Konzept jeweils auf den Plattformen Android, iOS und HTML5 entwickelt, um Kenntnisse über die Möglichkeiten der jeweiligen Plattformen für solch eine Anwendung zu erlangen. Diese Arbeit behandelt die Umsetzung der Version für Windows Phone.

In einer begleitenden Studie werden anschließend sowohl die vier Implementierungen, als auch die Auswirkungen verschiedenen Einstellungen verglichen.

1.2 Aufbau der Arbeit

In der nachfolgenden Arbeit werden zunächst in Kapitel 2 einige Grundlagen beleuchtet, die im späteren Verlauf der Arbeit aufgegriffen werden. Im Anschluss wird in Kapitel 3 der Implementierungsprozess der erstellten Anwendung beginnend beim Konzept bis zu ihrer fertigen Applikation nachvollzogen. Daraufhin wird in Kapitel 4 die im Anschluss an die Implementierung erfolgte begleitende Studie erläutert und auf die Ergebnisse

eingegangen. In der Zusammenfassung in Kapitel 5 werden zum Schluss die Entscheidungen während der Implementierungsphase und die Ergebnisse der Studie resümiert, bevor das Fazit in Kapitel 6 einen Ausblick auf die zukünftigen Möglichkeiten der Anwendung gewährt.

2 Grundlagen

In diesem Kapitel werden einige grundlegende Aspekte vermittelt, welche zum Verständnis der Arbeit beitragen. Zunächst wird das Phänomen des Tinnitus genauer beleuchtet und einige Behandlungsansätze vorgestellt. Daraufhin werden die für die räumliche Positionierung von Geräuschen wichtigen Grundlagen behandelt.

2.1 Tinnitus

Tinnitus bezeichnet das Wahrnehmen eines Geräusches ohne die nachweisbare Existenz einer externen Quelle. Das wahrgenommene Geräusch und dessen Lautstärke variiert dabei von Person zu Person und kann wie ein Fiepen, ein Rauschen oder ein gleichbleibender oder auch pulsierender Ton klingen. Dabei kann ein einmaliges Auftreten verschieden lange anhalten [30].

Man unterscheidet im allgemeinen zwischen zwei Arten von Tinnitus. Der objektive oder pseudo Tinnitus bezeichnet eine tatsächlich existente Geräuschquelle im Patienten, die auch durch andere Personen beispielsweise mit Hilfe eines Stetoskops wahrgenommen werden kann. So können die Geräusche beispielsweise durch eine Verkalkung der Halsschlagader oder durch einen Tumor verursacht werden. Diese Ursachen können beispielsweise durch einen operativen Eingriff oder Medikation erfolgreich behandelt werden [4].

Der subjektive Tinnitus, bei dem keine äußere Geräuschquelle existiert, ist jedoch die weitaus verbreitetere Form. Viele Menschen, vor allem in Industrieländern, erleben mindestens einmal in ihrem Leben das Phänomen eines Tinnitus. Die Gründe für dessen Auftreten sind vielzählig. Der häufigste Grund ist jedoch ein Gehörverlust, der durch laute Geräusche hervorgerufen wird. In den meisten Fällen verschwindet ein Tinnitus in einer relativ kurzen Zeitspanne, die zwischen einigen Sekunden und drei Monaten liegt. Ist dies der Fall, spricht man von einem akuten Tinnitus. Dauert die Wahrnehmung des non-existenten Geräusches jedoch länger als drei Monate, so spricht man von einem chronischen Tinnitus.

2 Grundlagen

Von einem dekompensierten Tinnitus spricht man sobald Folgeerkrankungen auftreten, die sich auf sämtliche Lebensbereiche auswirken. Dazu zählen Muskelverspannungen, Angstzustände, Konzentrations- und Schlafstörungen, sowie Depressionen [9].

Die genauen Ursachen der subjektiven Ohrgeräusche sind noch kaum erforscht. In den häufigsten Fällen entstehen sie wahrscheinlich nach einer Schädigung von Haarzellen im Innenohr durch Gehörverlust. Daraufhin erhält das Hörzentrum im Gehirn, der auditorische Cortex, fehlerhafte oder abgeschwächte Signale. Um das Signal zu verstehen, wird das Grundrauschen der Nervenzelle im Hörzentrum verstärkt [31]. Diese erhöhte Aktivität des auditorischen Cortex lässt sich bei Tinnituspatienten unter Laborbedingungen messen. Die entsprechenden Hirnregionen der Tinnituspatienten zeigen bei Stille einen Rückgang der Alphawellen, welche normalerweise ein Indikator für Entspannung einer Hirnregion darstellen [24].

Die Erforschung dieses Phänomens steckt noch in den Kinderschuhen und eine vollständige Heilung ist bislang nicht möglich. Es existieren nur unterschiedlichste Therapieansätze in verschiedenen Bereichen, um eine Linderung für den Patienten zu erzielen. Einige dieser Behandlungsmethoden werden nun vorgestellt [6].

Kognitive Verhaltenstherapie (KVT) Die KVT soll dem Patienten bewusst machen, dass er durch Umstrukturierung seines Denkens und Verhaltens besser mit einem Tinnitus leben kann. Es hat jedoch keinen Effekt auf die Ausprägungen des Tinnitus. Die KVT stellt das best evaluierte Therapieverfahren für Ohrgeräusche dar.

Hörgeräte Liegt die Tinnitusfrequenz unter 6 kHz, ist diese Art der Behandlung am effektivsten, da genau dieser Frequenzbereich durch Hörgeräte verbessert wird.

Tailor-Made Notched Music Training Hierbei wird aus Musikstücken der Frequenzbereich in dem der Tinnitus liegt herausgefiltert. Bei ersten Versuchen wurde die Lautstärke des Tinnitus der Studienteilnehmer signifikant gesenkt.

Auditorische Stimulation Da der Tinnitus meist in ruhigen Umgebungen als belastend empfunden wird, nutzt diese Behandlungsmethode eine gezielte auditorische Stimulation um den Tinnitus zu reduzieren.

Die in dieser Arbeit erstellte Anwendung macht sich das Prinzip der auditorischen Stimulation zunutze.

2.2 Räumliche Lokalisierung von Geräuschen

Die Wahrnehmung von Geräuschen und das Zuordnen der Richtung aus welcher sie kommen, ist ein komplexes Zusammenspiel mehrerer Faktoren. Das Gehör nutzt Informationen, die im an beiden Ohren ankommenden unterschiedlichen Schalldruck kodiert sind, um daraus einen räumlichen Höreindruck zu erzeugen.

Psychoakustische Experimente haben gezeigt, dass folgende im Schalldruck versteckte Hinweise zu einer räumlichen Lokalisierung von Geräuschen führen [32].

- Der Laufzeitunterschied, auch ITD (von engl. interaural time difference), bezeichnet die zeitliche Differenz in der die Schallwellen am linken und am rechten Ohr eintreffen. Dieser Unterschied ist besonders wichtig für Frequenzen die unter ca. 1,5 kHz liegen.
- Die Pegeldifferenz, auch ILD (von engl. interaural level difference), ist der Unterschied zwischen den an den Ohren ankommenden Schalldruck. Dieser ist besonders für Frequenzen über 1,5 kHz entscheidend. Die Pegeldifferenz, sowie der Laufzeitunterschied eines kurzen Geräusches auf horizontaler Ebene mit einem Abstand über 1m sind in Abbildung 2.1 abgebildet.
- Varianzen in den Schalldruckspektren, die an den Ohren ankommen. Abbildung 2.2 zeigt die Frequenzspektren von an den Ohren ankommenden Schallwellen bei einer Geräuschquelle auf horizontaler Ebene. Schallwellen breiten sich von ihrer Quelle durch den gesamten Raum aus und treffen dabei auf Oberflächen, wie beispielsweise Wände, von denen sie reflektiert, gebeugt und gestreut werden. Solch eine Streuung tritt auch am eigenen Körper auf. Für den Höreindruck wichtig sind dabei die Streuung am oberen Torso, dem Kopf und an den Ohrmuscheln [21]. Besonders die von der äußeren Ohrmuschel gestreuten hochfrequenten Töne über 5 oder 6 kHz, sind für die vertikale Lokalisierung, sowie für die Unterscheidung zwischen vorne und hinten ausschlaggebend [32]. Dies ist in Abbildung 2.2 gut zu erkennen.
- Dynamische Hinweise, die zum Beispiel durch Drehen des Kopfes, ITD und ILD verändern, dienen dem Gehirn ebenfalls ausschlaggebend zur Lokalisierung auf der vertikalen Ebene und der Unterscheidung von vorne und hinten.

Die Frequenz v_C unter welcher der Laufzeitunterschied dominant ist, kann mit Formel 2.1 berechnet werden. Hierbei bezeichnet c die Schallgeschwindigkeit und d den durchschnittlichen Durchmesser eines menschlichen Kopfes.

$$v_C = \frac{c}{d} \approx 1,5kHz, \quad \text{mit } c = 340 \frac{m}{s} \text{ und } d = 25cm. \quad (2.1)$$

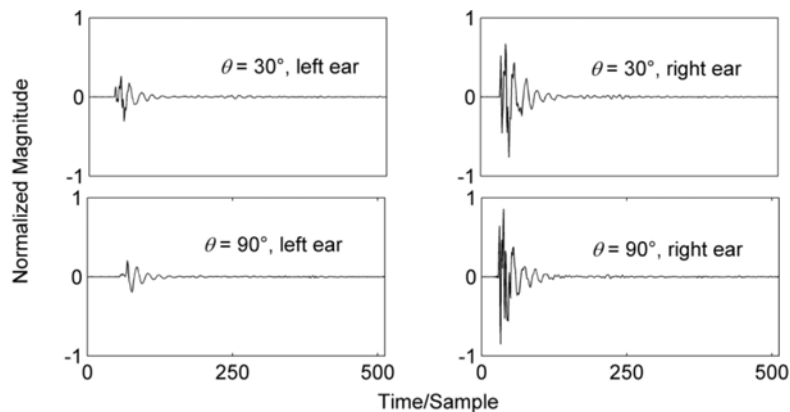


Abbildung 2.1: An beiden Ohren ankommende Schallwellen. Obere Reihe: 30 Grad Abweichung zur Front des Hörers. Untere Reihe: 90 Grad Abweichung [32]

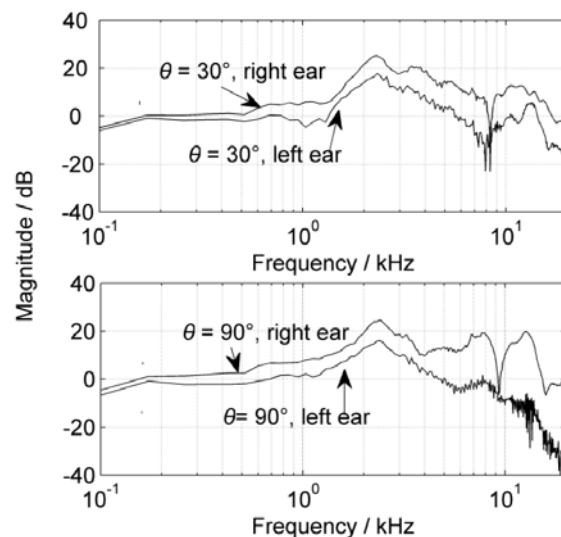


Abbildung 2.2: Frequenzspektren an beiden Ohren. Oben: 30 Grad Abweichung. Unten: 90 Grad Abweichung [32]

Die Laufzeitdifferenz und der Pegelunterschied genügen lediglich um den Standpunkt eines Geräusches einzugrenzen. Dies führt zum Phänomen des sogenannten "cone of confusion" (deutsch: Kegel der Verwirrung), wie in Abbildung 2.3 zu sehen ist. Darunter versteht man, dass die Richtung eines Geräusches entlang eines kegelförmigen Trichters nicht zu unterscheiden ist [5]. Es kann beispielsweise nicht unterschieden werden, ob sich ein Geräusch vor oder hinter dem Zuhörer befindet. Um die Richtung genauer zu bestimmen, nutzt das Gehirn die beschriebene Streuung der Schallwellen.

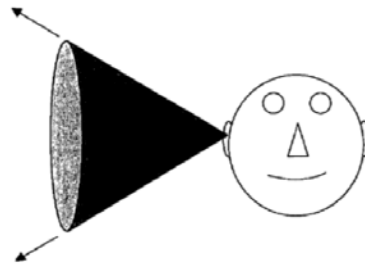


Abbildung 2.3: Cone of confusion [21]

2.2.1 Head-related Transfer Functions (HRTF)

Anders als bei der Wahrnehmung von natürlichen Geräuschen, die aus allen möglichen Richtungen kommen können, sind die Geräuschquellen bei der Tonwiedergabe mit Kopfhörern auf die beiden Lautsprecher direkt an den Ohren begrenzt. Dies hat den Effekt, dass die Schallwellen nicht, wie im vorhergehenden Abschnitt beschrieben, an Torso oder Kopf gestreut werden können. Jedoch spielt, wie in Abschnitt 2.2 bereits erläutert, die Streuung und Reflexion am Körper eine übergeordnete Rolle für das räumliche Hören. Durch das Wegfallen dieser Effekte scheint die Wahrnehmung des Geräusches durch Kopfhörer so, als läge die Geräuschquelle inmitten des Kopfes des Zuhörers.

Durch die Messung und Anwendung von head-related transfer functions (deutsch: kopf-bezogene Übertragungsfunktionen), kurz HRTF genannt, ist es möglich, diesen Eindruck zu umgehen und das empfundene Geräusch nach außen zu transportieren. Es kann also ein virtueller surround sound simuliert werden, bei dem versucht wird die "cone of confusion" zu umgehen. Mit Hilfe von im Labor gemessenen HRTFs, wie beispielsweise bereits in den Abbildungen 2.1 und 2.2 gesehen, wird versucht durch passende Fourier-Transformationen ein Stereo-Audiosignal in einen virtuellen surround sound umzuwandeln.

Im Folgenden wird die Messung und die mathematische Anwendung von HRTFs angesprochen.

Ermittlung von HRTFs

Um HRTFs zu ermitteln, wird oftmals großer Aufwand betrieben. Ein möglicher Versuchsaufbau ist in Abbildung 2.4 zu sehen. Zum Messen der Daten werden Dummies mit modellierten Ohrmuscheln mit integrierten Mikrofonen, oder alternativ, echte Men-

2 Grundlagen

schen, denen Mikrofone in den Ohren angebracht werden, verwendet. Es ist wichtig, dass der Versuchsraum möglichst schallisoliert ist, um unerwünschte Streuung von den Wänden oder dem Boden zu vermeiden [7].

Nachdem die Körper-, Kopf- und Ohrformen genau vermessen wurden, werden mit Hilfe einer Vielzahl von Lautsprechern, die um den Versuchskörper angebracht sind, Geräusche abgespielt. Mit Hilfe der Mikrofone, werden diese aus verschiedenen Raumrichtungen stammenden Geräusche auf ein Stereosignal abgebildet.



Abbildung 2.4: Versuchsaufbau zur Messung von HRTFs [32]

Berechnung von HRTFs

Unter der Annahme, sowohl Geräuschquelle, als auch Zuhörer seien feste Punkte, kann die Geräuschübertragung als linearer zeitunabhängiger Prozess betrachtet werden [32]. Basierend auf dem in Abbildung 2.5 gezeigtem System, sind die HRTFs die akustischen Übertragungsfunktionen H_L für das linke Ohr und H_R für das rechte Ohr definiert durch:

$$H_R(r, \theta, \phi, f, a) = \frac{P_R(r, \theta, \phi, f, a)}{P_0(r, f)}, \quad H_L(r, \theta, \phi, f, a) = \frac{P_L(r, \theta, \phi, f, a)}{P_0(r, f)}, \quad (2.2)$$

wobei P_L und P_R den Schalldruck am linken bzw. am rechten Ohr beschreiben, P_0 hingegen den Schalldruck am Mittelpunkt des Kopfes bezeichnet, wenn dieser nicht anwesend wäre. HRTFs sind Abhängig von der Frequenz f und der Position, (r, θ, ϕ) der Schallquelle, sowie vom Individuum a . Ist der Radius $r \lesssim 1m$, so sind HRTFs abhängig vom Radius. Andernfalls sind sie weitestgehend von diesem unabhängig [32].

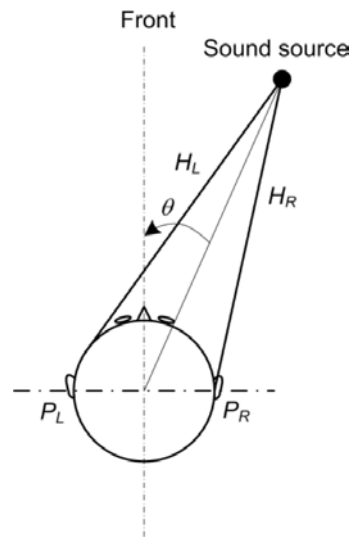


Abbildung 2.5: Schematische Darstellung der Übertragung eines Geräusches von einer Quelle zu den Ohren [32]

2.2.2 Auftretende Probleme mit HRTFs

Nachdem jeder Mensch unterschiedliche Körpermaße hat und sich auch die Form der Ohrmuscheln unterscheiden, ist es nicht möglich eine universale Formel für die HRTF zu erstellen [7]. Hersteller von derartigen Audiosystemen nutzen deshalb gemittelte Werte für die Ohr-, Kopf- und Torsoform. Diese Approximation funktioniert somit bei manchen Menschen besser und bei manchen jedoch weniger gut. So kommt es beispielsweise häufig zur Verwechslung von vorne und hinten, oder Geräusche klingen so, als lägen sie höher, als sie in Wirklichkeit sind [21].

3 Implementierung

In diesem Kapitel wird die Implementierungsphase der Anwendung genauer beleuchtet. Zunächst wird mit dem Konzept die Idee hinter der Applikation erläutert und die Anforderungen an die Anwendung analysiert. Im Anschluss wird genauer auf die für die Umsetzung des Konzepts verwendete Hardware und Software eingegangen. Daraufhin werden die wichtigsten Kernelemente während der Implementierung herausgestellt und die getroffenen Entscheidungen erklärt. Bevor zuletzt die Systemanforderungen für eine eventuelle Wiederverwendung der Anwendung geschildert werden, wird noch ein Vergleich zwischen dem ursprünglichen Konzept und der fertigen Anwendung gezogen.

3.1 Konzept der Anwendung

Dieser Abschnitt behandelt die Entwicklung für die mobile Plattform Windows Phone, sodass eine ähnliche Anwendung auf den meisten gängigen Mobilgeräten verfügbar ist.

Der Anwender soll sich während des Spiels stark auf die auditorische Stimulation, die er durch die Anwendung erfährt, konzentrieren. Ziel ist es, den Benutzer somit ausreichend abzulenken, um den Tinnitus nicht mehr aktiv wahrzunehmen. Dies soll dem Benutzer eine, zumindest kurzfristige, Linderung seines Tinnitusleidens verschaffen.

Damit sich der Benutzer sehr stark auf ein anderes Geräusch als seinen Tinnitus konzentrieren muss, verfolgt die Anwendung die Idee, dass der Nutzer ein Geräusch orten soll. Der Benutzer hört über die Kopfhörer ein Geräusch, welches zufällig an einer Position um ihn herum erklingt. Nun hat er die Aufgabe das Geräusch zu orten und sich mit seinem Smartphone in die Richtung zu drehen, aus welcher das Geräusch ertönt. Dabei erhält der Benutzer kontinuierlich auditorisches Feedback über seine aktuelle Position in Relation zum Geräusch. Hört der Benutzer beispielsweise ein Geräusch rechts von ihm, dreht er sich mit seinem Smartphone in der Hand nach rechts und hört das Geräusch nun vor sich.

Um einen Anreiz zum öfteren Spielen zu schaffen, bei dem der Anwender auch Spaß hat, ist der Aspekt der Gamifizierung sehr wichtig. Hierbei sollen beim User Anreize

3 Implementierung

geschaffen werden, die Anwendungen öfter ausführen zu wollen [2]. Um das eben erläuterte Konzept als Spiel umzusetzen, verfolgt die Applikation einen Safari- oder Naturforscheransatz. Der Spieler muss mit seinem Smartphone Fotos von Tieren schießen, die er nicht sehen, sondern nur hören kann. Dazu soll der Benutzer sein Smartphone so vor sich halten, als wolle er ein Foto aufnehmen und dreht sich anschließend zu dem Tiergeräusch, welches er an einer Position um sich herum hört. Hat der Spieler das Geräusch gefunden, sieht er anschließend auf dem Display seines Smartphones ein Foto des gehörten Tieres. Das gibt dem User im Nachhinein auch die Möglichkeit visuell zu überprüfen, wie weit er von der Geräuschquelle entfernt war. Der Benutzer macht sozusagen ein Foto seiner virtuellen Umgebung. Abbildung 3.1 soll das Konzept der virtuellen Umgebung verdeutlichen.

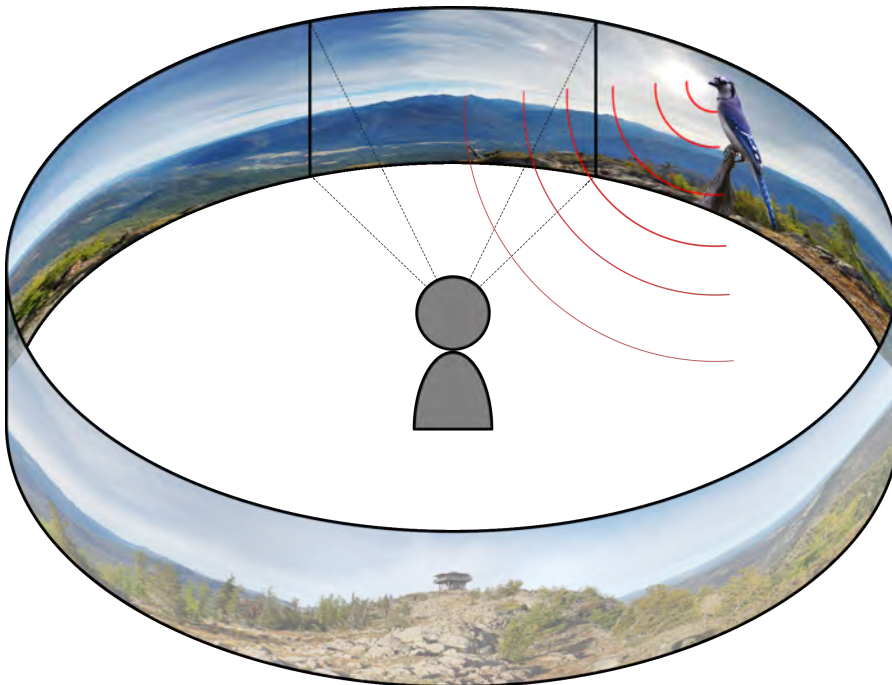


Abbildung 3.1: Sichtbarer Ausschnitt einer virtuellen Umgebung

Um dem Rahmen einer Bachelorarbeit nicht zu überschreiten, wird im Nachfolgenden beschrieben, wie ein Prototyp dieses Konzeptes entworfen und implementiert wird. Bei diesem soll der Benutzer die Möglichkeit haben ein oder zwei unterschiedliche Tiergeräusche gleichzeitig zu hören und nacheinander suchen zu müssen. Ebenso kann das Spiel mit oder ohne ein zusätzliches Hintergrundgeräusch gespielt werden. Diese Parameter werden aufgrund der anschließend folgenden Studie gewählt, welche genauer im Abschnitt 4 besprochen wird.

3.2 Anforderungsanalyse

Im folgenden Abschnitt werden die Anforderungen an die zu entwickelnde Anwendung analysiert. Dabei wird sowohl auf funktionale, wie auch auf nicht-funktionale Anforderungen eingegangen. Dabei wird auch der optische Stil der zu entwickelnden Anwendung festgelegt.

3.2.1 Funktionale Anforderungen

Die funktionalen Anforderungen bezeichnen Funktionen, welche die Anwendung ausführen können muss. Im nachfolgenden sind diese essentiellen Funktionen aufgelistet.

FA#1: Räumliche Positionierung von Geräuschen: Das vom Spieler zu suchende Geräusch befindet sich in einem virtuellen Raum. Dabei liegt die Geräuschquelle auf einem Punkt einer Kreisbahn, deren Zentrum der Spieler bildet. Dreht sich der Spieler, verändern sich auch die Positionen der Geräusche relativ zum Spieler. Dies erfolgt in Echtzeit.

FA#2: Geräusche werden leiser, wenn sie hinter dem Spieler liegen: Bei der Ortung des Geräusches im virtuellen Raum, spielt nicht nur die Unterscheidung von links und rechts eine Rolle. Der Spieler muss ebenso gut unterscheiden können, ob sich das Geräusch vor oder hinter ihm befindet. Da das 'virtuelle Ohr' omnidirektional ist, hört es nach vorne genauso gut, wie nach hinten. Wie im Abschnitt 2.2.2 bereits erwähnt, treten daher häufig Probleme bei der schwierigen Unterscheidung zwischen vorne und hinten auf. Um diese Unterscheidung zu erleichtern, wird das Geräusch leiser, je weiter es hinter dem Spieler liegt.

FA#3: Ein oder zwei Geräusche gleichzeitig: Der Spieler hat die Wahl, ob ein oder zwei Geräusche gleichzeitig bei der Suche abgespielt werden. Dabei unterscheiden sich sowohl die Geräusche, als auch ihre Position eindeutig.

FA#4: Verschiedene Geräusche wählen: Vor Beginn des Spiels können die zu suchenden Geräusche aus einer Auswahl gewählt werden. Dies kann entweder vom Spieler selbst entschieden werden, oder von der Anwendung als Teil verschiedener Levels geschehen.

FA#5: Visuelles Feedback: Auf der Ergebnisseite bekommt der Spieler ein Bild präsentiert. Dieses zeigt wie bereits in Abbildung 3.1 gezeigt, einen Ausschnitt aus der virtuellen Umgebung um den Benutzer.

FA#6: Anzeige der Abweichung in Grad: Zusätzlich zu dem Bild, bekommt der Benutzer noch seine genaue Abweichung zur Geräuschquelle angezeigt. Dem Spieler wird eine Gradanzahl mit der zugehörigen Seite auf der er entfernt war angezeigt. Beispielsweise "8 Grad zu weit rechts".

FA#7: Logfiles: Für die Auswertung der Studie, siehe Kapitel 4, ist es wichtig, dass die vom Spieler produzierten Daten gespeichert werden [28]. Den Logfiles sollen im Nachhinein alle für die Studie relevanten Daten, die während des Spiels anfallen, zu entnehmen sein.

3.2.2 Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen bezeichnen die Anforderungen an die Anwendung, welche über die funktionalen Anforderungen hinausgehen und stellen somit Anspruch an die Qualität der Anwendung. Im nachfolgenden sind diese aufgelistet.

NFA#1: Intuitive Steuerung: Die Navigation durch die Menüs, sowie der spielerische Aspekt der Anwendung sollen selbsterklärend und dabei intuitiv zu steuern sein.

NFA#2: Schlichtes Design: Das Design soll einfach gehalten werden und dabei die Vorgaben des im nachfolgenden Abschnitts 3.2.3 festgeschriebenen Style guides erfüllen.

3.2.3 Styleguide

Das Erscheinungsbild der mobilen Anwendung soll übersichtlich und strukturiert wirken. Dabei soll auch das allgemeine Look & Feel von Windows Phone Anwendungen beibehalten werden. Deswegen werden sehr viele Designvorgaben der Windows Apps übernommen. Im nachfolgenden wird kurz auf die wichtigsten Designaspekte eingegangen.

Layout

Die Anwendung wird in Anlehnung an Fotografie mit dem Smartphone im Querformat gestaltet. Da das der Benutzer das Smartphone sicher vor sich halten können muss, werden alle Elemente werden so angeordnet, dass die mobile Anwendung problemlos mit einer Hand bedient werden kann.

Verwendete Farben in Menüs

In den Menüs der Anwendung wird auf eine schlichte Gestaltung Wert gelegt. Ebenso wie die Menüführung, sollen auch die verwendeten Farben den User nicht ablenken und eine intuitive Bedienung unterstützen. Abbildung 3.2 zeigt das in den Menüs der mobilen Anwendung verwendete Farbschema.

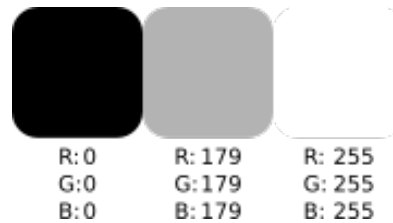


Abbildung 3.2: Verwendetes Farbschema

Wie es typisch für Windows Phone Apps, wie auch für die Menüs des Gerätes selbst ist, werden die Menüs mit weißer Schrift auf schwarzem Hintergrund gestaltet. Buttons und andere Auswahlmöglichkeiten werden nach Windows 8.1 Vorgaben transparent mit weißem Rand und eventuell weißer Schrift dargestellt [11].

Texteingabefelder und sonstige Objekte werden in grau angezeigt.

Verwendete Schriftarten

Es werden die für Windows Phone Anwendungen standardmäßigen Schriftarten und Größen gewählt. Diese sind wie nachfolgend beschrieben vorgegeben.

Überschriften: Segoe WP SemiLight, 54 pt

Fließtext: Segoe WP, 15 pt

Buttontexte: Segoe WP, 17 pt

3.2.4 Weitere Features

Neben den oben beschriebenen Kernelementen der Applikation, wurden noch zusätzliche Features in die Planung der Anwendung mit einbezogen. Im folgenden sind diese aufgelistet.

Hintergrundbild auswählen: Ebenso wie die verschiedenen Geräusche, kann auch das Hintergrundbild, welches auf der Ergebnisseite angezeigt wird, gewechselt werden. Dies kann ebenfalls Bestandteil einer Levelauswahl sein.

Filter auf Geräusche anwenden: Zusätzlich zu der Linderung des Tinnitus durch Ausblenden, kann durch einen Passfilter die Frequenz des Tinnitus aus den zu suchenden Geräuschen ausgefiltert werden. Wie bereits im Abschnitt über die Tinnitusbehandlung 2.1 erwähnt, muss für diese Behandlung zuvor jedoch die genaue Frequenz des Tinnitus bekannt sein.

Steuerung ausschließlich über den Touchscreen: Besitzt ein Smartphone keinen der zum Spielen nötigen Sensoren, kann das Spiel ausschließlich über den Touchscreen gesteuert werden. Dabei wird die Drehung des Spielers über zwei Pfeile an den Bildschirmrändern vorgenommen.

3.2.5 Bewertung und Einteilung der Features

Um die Funktion der fertigen Anwendung sicherzustellen, werden die Prioritäten klar auf die wichtigsten Features verteilt. Um dies übersichtlich zu gestalten, wird allen angesprochenen Funktionen in Tabelle 3.1 auf Seite 19 eine Kategorie zugewiesen, die die Relevanz der jeweiligen Funktion angibt. Die Kategorien sind wie folgt definiert.

muss: Das Feature ist eine Kernfunktion und muss in der finalen Version implementiert sein.

soll: Das Feature soll in der fertigen Version enthalten sein. Es ist keine Funktion ohne die die Anwendung nicht funktioniert, trägt aber maßgebend zu dieser bei.

kann: Das Feature kann in der finalen Version enthalten sein, wird aber weggelassen, sollten andere Details wichtiger sein.

3.2.6 Anforderungen an das verwendete Smartphone

An das verwendete Smartphone werden durch das Spielkonzept auch gewisse Anforderungen gestellt. Die Applikation wurde für Windows Phone 8 entwickelt, um somit auf möglichst vielen Geräten problemfrei ausgeführt werden zu können. Somit muss das verwendete Smartphone ebenfalls mindestens Windows Phone 8 als Betriebssystem nutzen. Die Anwendung ist natürlich aufwärtskompatibel. So läuft das Programm ebenfalls auf Windows Phone 8.1 und - soweit zum Zeitpunkt dieser Arbeit bekannt - auch auf Windows 10.

Da die Steuerung der Anwendung zum Teil durch Drehung des Gerätes konzipiert ist, sind gewisse Sensoren eine Mindestanforderung, da ohne diese eine korrekte Anwen-

Feature	muss	soll	kann
Räumliche Positionierung von Tönen	x		
Leiser, wenn Ton hinten		x	
Auswahl ob ein oder zwei Geräusche	x		
Mehrere auswählbare Geräusche			x
Hintergrundbild wählbar			x
Filter auf Töne anwenden			x
Nur Touchscreen-Steuerung			x
Visuelles Feedback	x		
Anzeige der Abweichung in Grad		x	
Logfiles anlegen	x		

Tabelle 3.1: Muss-, Soll-, Kann-Klassifikation der Features

ung nicht garantiert werden kann. Das Gerät muss mindestens einen integrierten Kompass besitzen. Ein Beschleunigungssensor trägt zusätzlich dazu bei, die Präzision des Kompasses zu verbessern. Für die beste Benutzererfahrung ist ein Gyroskop von besonderer Bedeutung. Hierdurch lässt sich die Lage des Smartphones erfassen, was in Kombination mit dem Kompass und dem Beschleunigungssensor zu einer exakten Bestimmung der Lage des Gerätes beiträgt. In Abschnitt 3.5.2 wird dieser Sachverhalt noch genauer erläutert.

Die Anwendung funktioniert nur dann korrekt, wenn sie in Verbindung mit Kopfhörern ausgeführt wird. Eine räumliche Wiedergabe von Geräuschen ist mit den im Smartphone integrierten Lautsprechern nicht möglich.

3.3 Konzept des User Interface

In der Anforderungsanalyse wurde als nicht-funktionale Anforderung eine intuitive Bedienung gefordert. Um dieser Forderung bestmöglich gerecht zu werden, ist die Steuerung der Anwendung an bereits bekannte Tätigkeiten mit dem Smartphone angelehnt. Hierbei liegt der Fokus auf dem Fotografieren via Touchscreen-Interaktion. Das Foto wird aufgenommen sobald der User eine Schaltfläche auf dem Bildschirm berührt. Um diese Immersion der Fotografie-Anwendung aufrecht zu erhalten, wird die gesamte mobile Anwendung im Querformat gestaltet.

3 Implementierung

Dies ermöglicht ebenfalls eine große Darstellung des visuellen Feedbacks, nachdem im Spiel ein Foto gemacht wurde. Um eine optimale Usability zu gewährleisten, soll die Applikation so schlicht und komfortabel wie möglich gestaltet sein. Deswegen wird die Anwendung in drei einzelne, miteinander verlinkte Seiten aufgeteilt, die den Benutzer einfach und intuitiv durch die Anwendung führen. Diese drei Seiten werden nun in Bezug auf ihre benötigten UI Elemente erläutert.

Hauptmenü

Das Hauptmenü wird angezeigt, wenn die Anwendung gestartet und bevor ein neues Spiel begonnen wird. Hier kann der Benutzer die Einstellungen der nächsten Spielrunde festlegen. Als mögliche Spielvarianten kann der Benutzer zwischen einem oder zwei zu suchenden Geräuschquellen wählen. Ob während der Suche ein Hintergrundgeräusch wiedergegeben werden soll, oder nicht, kann der User hier ebenfalls festlegen. Die verschiedenen Einstellungen sollen dabei so übersichtlich wie möglich präsentiert werden, um die Auswahl einfach und schnell zu gestalten. Zu viele Details bei den Einstellungen würden die Motivation des Benutzers mindern, das Spiel zu starten (siehe Styleguide, Abschnitt 3.2.3).

Spiel

Während der eigentlichen Spielphase soll sich der Benutzer auf den auditorischen Input konzentrieren. Demnach sollte das User Interface nicht ablenken, da der Benutzer nicht auf den Bildschirm schauen muss. Somit wird auf die Anzeige der aktuellen Drehposition verzichtet, da eine sich laufend verändernde Anzeige ablenkend ist. Die räumliche Orientierung in Bezug auf die Geräuschquellen soll ausschließlich über das Gehör erfolgen. Dem Benutzer soll lediglich eine große Schaltfläche zur Verfügung stehen, mit deren Hilfe er ein 'Foto aufnimmt', indem er den Bildschirm berührt.

Ergebnisse

Auf der Ergebnisseite erhält der Benutzer visuelles Feedback zu seinem 'Foto', welches er während der Spielphase durch das Tippen auf dem Bildschirm aufgenommen hat. Dazu soll als Hintergrund über den ganzen Bildschirm ein Ausschnitt eines 360-Grad-Panoramabildes angezeigt werden. Je nachdem in welcher Position der Spieler gerichtet

war, unterscheidet sich dieser Ausschnitt. Auf diesem Bildausschnitt wird, sofern der Benutzer an der richtigen Stelle getippt hat, ein zu der Geräuschquelle passendes Bild angezeigt. Je genauer der Spieler die Position der Geräuschquelle getroffen hat, desto zentrierter erscheint die Geräuschquelle, die in diesem Fall ein Tier darstellt. Zusätzlich soll dem Benutzer am unteren Bildschirmrand eine Anzeige zur Verfügung stehen, aus der er seine Abweichung in Grad und Richtung entnehmen kann. Beispielsweise "7 Grad zu weit links".

Außerdem wird ein Button benötigt, mit welchem der Benutzer je nach Einstellung das aktuelle Spiel fortsetzen oder zurück in das Hauptmenü gelangen kann.

3.3.1 User Interaktionen

Im nachfolgenden Abschnitt werden die oben geforderten UI-Elemente der einzelnen Seiten weiter spezifiziert und in Hinsicht auf die Nutzerinteraktionen betrachtet.

Hauptmenü

Wie bereits in Abschnitt 3.3 beschrieben, kann der Benutzer im Hauptmenü nachfolgende Entscheidungen treffen. Er kann zwischen einer und zwei Geräuschquellen wählen und zudem noch bestimmen ob ein Hintergrundgeräusch wiedergegeben werden soll. Für die Auswahl der Anzahl zu suchender Geräusche sind zwei Radiobuttons eine geeignete Wahl.

Zwar wäre auch eine Checkbox denkbar, da es nur zwei mögliche Optionen gibt, jedoch ist die explizite Auswahl, mit je einem eigenen Radiobutton wesentlich intuitiver. Des Weiteren kann die Auswahl auf diese Weise in Zukunft leicht erweitert werden, ohne das Menü umstellen zu müssen. Es wäre beispielsweise denkbar, dass der Benutzer in Zukunft auch drei oder mehr verschiedenen Geräusche wählen kann.

Für das Ein- oder Ausschalten des Hintergrundgeräusches ist eine Checkbox ausreichend. Um das Spiel zu starten genügt ein Button, der den Benutzer direkt auf die Spielseite weiterleitet.

Spiel

Die Benutzerinteraktionen während der Spielphase beschränken sich auf zwei Aktionen. Zum einen das Drehen des Spielers mit seinem Smartphone, welches er in der Hand

3 Implementierung

hält um die Geräuschquellen zu orten. Zum anderen das Tippen auf den Auslöser-Button auf dem Bildschirm, um seine Position zu sichern und in Folge auf die Ergebnis-seite weitergeleitet zu werden.

Ergebnisse

Zur Interaktion mit der Anwendung steht dem Benutzer auf der Ergebnisseite nur ein Button zur Verfügung. Je nach Auswahl im Hauptmenü, kann der Benutzer mit diesem Button sein aktuelles Spiel fortsetzen oder zum Hauptmenü zurückkehren. Hat der Benutzer zwei Geräuschquellen ausgewählt und aktuell die erste Quelle gefunden, kann der Benutzer mit Hilfe des Buttons zur Spielseite zurück kehren, um dort das zweite Geräusch zu suchen. Hat der Spieler das zweite Geräusch gefunden, oder nur eine zu suchende Geräuschquelle gewählt, leitet ihn der Button zum Hauptmenü weiter, um dort ein neues Spiel zu starten und die Einstellungen erneut festzulegen.

Bei der Positionierung des Buttons ist darauf zu achten, dass sich dieser in der selben Lage, wie der Start-Button des Hauptmenüs befindet. Hierdurch wird ein schnelles Wiederholen der Spielrunde ermöglicht, ohne dass der Benutzer seinen Finger bewegen muss.

3.4 Verwendete Hardware und Entwicklertools

Um die Vorgaben des oben beschriebenen Konzepts umzusetzen, wurde spezielle Entwicklungssoftware für Windows genutzt. Diese, sowie die verwendete Hardware, werden nun besprochen.

3.4.1 Hardware

Für die Entwicklung des Programmes wurde ein PC mit dem Betriebssystem Windows 8.1 verwendet, wie es in Abbildung 3.3 zu sehen ist. Als Smartphone wurde sowohl für den Entwicklungsprozess als auch für die Durchführung der Studie ein Nokia Lumia 830 mit Windows Version 8.1 benutzt.



Abbildung 3.3: Nokia Lumia 830 (rechts) [18]

3.4.2 Tools

Als primäres Entwicklertool wurde "Visual Studio 2010 Express for Windows Phone" verwendet. Dabei handelt es sich um eine freie Community Version von Visual Studio 2010 mit integriertem Windows Phone 8 SDK (Software Development Kit). Im Endstadium der Entwicklung wurde auf Visual Studio 2015 Community gewechselt, welches ebenfalls das SDK für Windows Phone 8, neben höherer Versionen beinhaltet. Diese neuere Version ist vollständig kompatibel mit der zuvor benutzten Version, wird jedoch im Gegensatz zur Version Express 2010 noch von Windows gewartet.

Zum Testen und Debuggen wurde die Anwendung direkt auf dem Nokia Lumia 830, welches via USB-Kabel an den PC angeschlossen war, gestartet. Dies hatte sowohl den Vorteil des schnellen Debuggens ohne zuvor einen Emulator starten zu müssen, als auch die Möglichkeit das Gyroskop zu testen. Diese Option bietet der Emulator hingegen nicht.

Die verwendeten Tierbilder wurden mit Hilfe von Gimp freigestellt.

3.5 Herausforderungen und Probleme während der Implementierung

Während der Implementierung traten zunächst einige Probleme auf. Diese und deren Lösungen werden im nachfolgenden Kapitel beschrieben.

3 Implementierung

Anschließend werden die wichtigsten Kernelemente der Anwendung erläutert und deren Umsetzung gezeigt.

3.5.1 Aufgetretene Probleme

Das größte Problem stellte das zunächst verwendete Smartphone Nokia Lumia 730 dar, da dieses nicht über ein eingebautes Gyroskop verfügt [17]. Das Problem fiel bei einem ersten vertikalen Prototypen zum Testen der Rotation auf. Der Bildausschnitt, der sich bei Drehen des Smartphones gleichmäßig bewegen sollte, bewegte sich nur sprunghaft und sehr unruhig. Auch der Versuch die Bildsprünge über Mittelung der aktuellsten Sensordaten zu vermeiden, brachte nur mäßige Resultate. Die Lösung des Problems war die Verwendung eines mit Gyroskop ausgestatteten Smartphones, mit dem der Rotationstest problemlos funktionierte.

Ein weiteres Problem stellte das Laden der Bilder in die mobile Anwendung dar, da es mit den im .NET-Framework bereitgestellten Methoden nicht ohne weiteres möglich ist, die Maße und somit das Seitenverhältnis der Bilder während der Laufzeit zu erkennen. Das Seitenverhältnis wird jedoch zur Skalierung der Bilder benötigt. Das Problem wurde gelöst, indem die verwendeten, zunächst auf das jeweilige Tier zugeschnittenen, Tierbilder in einem Bild mit dem Seitenverhältnis von 1:1 im Assets-Ordner vorliegen. Somit spielt das Seitenverhältnis des Bildes keine Rolle mehr und die Grafik wird zur Laufzeit richtig skaliert.

3.5.2 Umsetzung der wichtigsten Kernelemente

Im nachfolgenden Abschnitt, werden die bedeutendsten Punkte zur Funktionsweise der mobilen Anwendung erläutert.

Rotationserkennung

Um die wichtigste Funktion der Steuerung aus dem Konzept umzusetzen, wurde auf eine Klasse der Windows Phone Sensoren Programmierschnittstelle (API) zurückgegriffen. Die sogenannte "Combined Motion" Klasse kombiniert die rohen Daten der einzelnen ihr zur Verfügung stehenden Sensoren und erlaubt der Anwendung direkt auf die Orientierung des Gerätes zuzugreifen [19]. Die räumliche Lage wird mit Hilfe von Roll-Nick-Gier-Winkeln (englisch: roll-pitch-yaw angle) angegeben.

3.5 Herausforderungen und Probleme während der Implementierung

Mit dieser Methode wird die Orientierung eines Objektes durch drei Winkel angegeben. Je ein Winkel bezeichnet dabei die Rotation entlang einer Körperachse des Objekts. Die Körperachsen sind in Abbildung 3.4 anhand eines Flugzeugs verdeutlicht. Die einzelnen Achsen bezeichnen dabei folgendes:

Rollen: Dies bezeichnet die Drehung um die Längsachse des Objekts. Auf die Zeichnung bezogen würde sich das Flugzeug rollen.

Nicken: Bezeichnung für die Drehung um die Querachse des Objekts. In der Zeichnung würde sich das Flugzeug nach oben oder unten zeigen, bzw. nicken.

Gieren: Dies ist die Rotation um die Vertikalachse. Bezogen auf die Zeichnung würde sich das Flugzeug im Ganzen um den Mittelpunkt quer drehen. Bei Fahrzeugen wird diese Rotation auch "Schlingern" genannt.

Die Applikation soll auf die Drehung des Smartphones um die Achse des stehenden Spielers reagieren. Demnach verwendet die Anwendung den Yaw- bzw. Gierungswinkel, der ihr von der "Combined Motion" Klasse zur Verfügung gestellt wird.

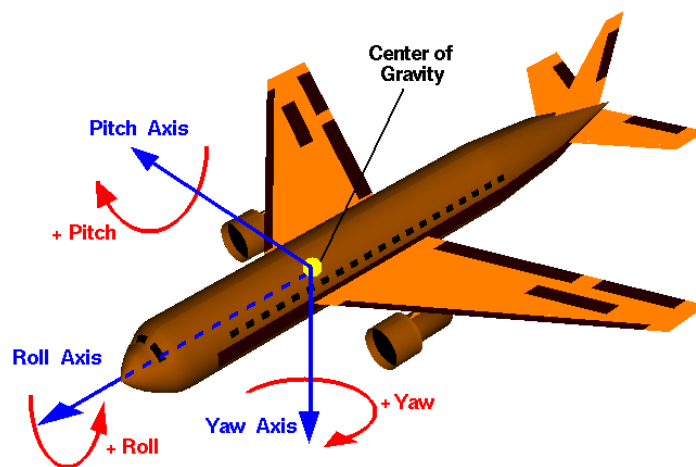


Abbildung 3.4: Veranschaulichung der Achsen eines Körpers [20]

Die "Combined Motion" Klasse benötigt um zu funktionieren mindestens einen Kompass- (auch Magnetometer-Sensor genannt) und einen Beschleunigungssensor. Jedoch sind die mit diesen beiden Sensoren zurückgelieferten Werte nicht sehr präzise und bieten keine gute Spielerfahrung. Genauere Werte liefert die "Combined Motion" Klasse, wenn ihr zusätzlich ein Gyroskop zur Verfügung steht [14].

Positionierung der virtuellen Geräusche

Um die Geräusche um den Spieler herum bewegen zu können, wurde ein virtuelles Koordinatensystem aufgespannt. Im Ursprung des Koordinatensystems wird eine Instanz der `AudioListener` Klasse aus dem Xna-Framework erzeugt. Dieses Objekt ist in der Lage Geräusche, die von Instanzen der Klasse `AudioEmitter` abgespielt werden, zu empfangen [13]. Jedes zu suchende Geräusch besteht aus einer Geräuschquelle, welche eine Instanz der `AudioEmitter` Klasse ist, die eine `AudioEffectInstance` abspielt.

Den Geräuschquellen wird bei ihrer Instanziierung eine Startposition in Form eines Winkels zugewiesen. Dieser Winkel gibt an, um wie viel Grad die Geräuschquelle in Relation zur Anfangsblickrichtung des Spielers auf einer Kreisbahn mit festem Radius im Uhrzeigersinn versetzt ist. Abbildung 3.5 auf Seite 27 verdeutlicht die Vorstellung des Koordinatensystems mit einer Kreisbahn um den Spieler, wobei die Blickrichtung des Anwenders immer die positive Z-Achse ist.

Um ein Geräusch zu Orten dreht sich der Benutzer mit seinem Smartphone in der Hand um die eigene Achse. Da im Koordinatensystem die positive Z-Achse der Blickrichtung des Spielers entspricht, bleibt der Spieler virtuell stehen und die Geräusche bewegen sich um ihn. Dreht sich der Spieler beispielsweise zu einem Geräusch zu seiner Rechten, bewegt sich die Geräuschquelle der Rotation entsprechend entlang der Kreisbahn gegen den Uhrzeigersinn. Die Bewegung der Geräusche entlang der Kreisbahn wird durch einfache Trigonometrische Funktionen gelöst. Die Koordinaten (x, y) des Geräusches werden grundlegend mit Formel 3.1 berechnet.

$$x = r \cdot \cos(\alpha), \quad y = r \cdot \sin(\alpha) \quad (3.1)$$

Wobei r der Radius der festgelegten Kreisbahn ist und α den Winkel bezeichnet, den das Geräusch relativ zur Blickrichtung des Benutzers besitzt.

Als Algorithmus wurde dieses Prinzip wie in Code 3.1 gezeigt umgesetzt. Sobald sich die Lage des mobilen Gerätes verändert und dies von der Motion API erkannt wurde, wird der aktuelle Drehwinkel "angle" in der Methode "CurrentValueChanged" ermittelt. Anschließend werden die Positionen der Geräusch-Emitter neu berechnet und festgesetzt. Dazu wird der aktuelle Drehwinkel zusammen mit der Anfangsposition der Geräuschquelle der Methode "currentEmitterPosition" übergeben. Diese berechnet nach dem in Formel 3.1 gezeigten Prinzip einen Positionsvektor. Nachdem der Geräuschquelle ihre neue Position zugewiesen wurde, werden die für den surround

sound wichtigen Schritte eingeleitet. Diese werden im nächsten Abschnitt genauer besprochen.

```
1 private void CurrentValueChanged(MotionReading e){
2     if (motion.IsDataValid){
3         angle = MathHelper.ToDegrees(e.Attitude.Yaw);
4         soundEmitter1.Position = currentEmitterPosition(angle, sound1.position);
5         adjustVolume(soundEmitter1, soundEffectInstance1);
6         soundEffectInstance1.Apply3D(soundListener, soundEmitter1);
7         [...]
8     }
9 }
10 public Vector3 currentEmitterPosition(float angle, float emitterAngle){
11     float winkel = MathHelper.ToRadians(emitterAngle - angle + 90);
12     float x = radius * (float) Math.Cos(winkel);
13     float y = radius * (float) Math.Sin(winkel);
14     Vector3 pos = new Vector3(x, 0, y);
15     return pos;
16 }
```

Code 3.1: Methoden zum Setzen der Emitter

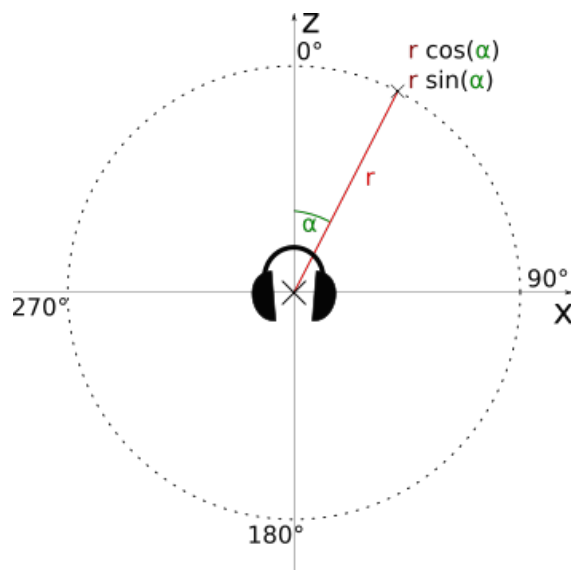


Abbildung 3.5: Veranschaulichung des Koordinatensystems zur Positionierung der Geräusche

Virtueller Surround Sound

Um die zu suchenden Geräusche auch räumlich auf den Kopfhörern des Spielers wiederzugeben, Geräusche links vom Spieler, also auf dem linken Ohr lauter zu hören sind, als

3 Implementierung

auch dem rechten, wurde eine Methode der AudioEffectInstance-Klasse des Microsoft.xna-Frameworks verwendet. Die Methode "Apply3D" wird nach einer Positionsänderung einer AudioEffectInstance auf dieser aufgerufen. Dabei werden der Methode sowohl die AudioListener-Instanz auf dem Ursprung des Koordinatensystems, als auch eine oder mehrere AudioEmitter-Objekte übergeben [12]. Die Methode errechnet anschließend mit Hilfe einer HRTF (siehe Abschnitt 2.2.1) ein Stereo-Geräusch, welches ein im Raum lokalisiertes Geräusch simuliert.

Die Apply3D-Methode gibt zwar die Geräusche in Stereo wieder, jedoch ist, wie bereits in Abschnitt 3.2.1 beschrieben, nicht unterscheidbar, ob sich das Geräusch vor oder hinter dem Zuhörer befindet. Wie in der Anforderungsanalyse gefordert, soll diese Unterscheidung ermöglicht werden, indem das Geräusch leiser wird, je weiter es sich hinter dem Spieler befindet. Um dies in der Implementierung zu bewerkstelligen, wird die Hilfsmethode "adjustVolume", wie im Codefragment Code 3.2 gezeigt, bei jeder Positionsänderung eines Geräusches für dieses aufgerufen. Dabei geht die Methode so vor, dass zunächst das Vorzeichen der Z-Komponente der Koordinate einer Geräuschquelle geprüft wird.

Wie in Abbildung 3.5 zu sehen ist, befindet sich das Geräusch vor dem User, falls die Z-Komponente positiv ist und hinter dem Benutzer, falls sie negativ ist. Liegt die Geräuschquelle hinter dem Spieler, wird über den Quotienten aus z-Koordinate und Radius - somit dem maximalen Abstand - ein Korrekturfaktor berechnet, der angibt wie weit sich das Geräusch hinter dem Spieler befindet. Dieser Korrekturfaktor liegt im Intervall $(0, 1]$, wobei Werte nahe 0 neben dem Spieler und 1 genau hinter dem Spieler bedeuten. Der Faktor wird von der maximalen Lautstärke 1 subtrahiert.

```
1 private void adjustVolume
2 (AudioEmitter emitter, SoundEffectInstance instance){
3     float volume;
4     if (emitter.Position.Z < 0){
5         volume = (float)(1 + (emitter.Position.Z / radius));
6     } else{
7         volume = 1;
8     }
9     instance.Volume = volume;
10 }
```

Code 3.2: Hilfsmethode "adjustVolume"

Mindestabstand zwischen zwei Geräuschen

Beim Starten der Spielphase werden die benötigten Geräuschquellen initiiert. Dazu werden je nach getätigten Einstellungen ein oder zwei Instanzen der Klasse "Soundquelle" instanziiert. Die Objekte der Klasse Soundquelle besitzen die in Code 3.3 gezeigten Variablen.

```
1 public string name { get; set; }
2 public float position { get; set; }
3 public SoundEffect sound { get; set; }
4 public Image image { get; set; }
```

Code 3.3: Objektvariablen des "Soundquelle"-Objekts

Dabei gibt "name" den Namen des zu suchenden Tieres an. Dieser wird auch während der Suche als Aufgabe angezeigt. "Position" legt, wie oben bereits beschrieben, die Startposition relativ zur Blickrichtung des Spielers fest. "Sound" und "image" bezeichnen das vom Spieler während der Spielphase gehörte Geräusch, beziehungsweise das auf der Ergebnisseite angezeigte Bild.

Die Werte der Anfangsposition werden zufallsgeneriert. Jedoch liefern Zufallsgeneratoren gerade bei mehrmaligem Aufruf hintereinander meist ähnliche Werte. Um zu verhindern, dass die beiden zu suchenden Geräusche direkt nebeneinander liegen und damit den Spieler nur verwirren würden, wird bei der Generation der Startposition der zweiten Geräuschquelle darauf geachtet, dass diese mindestens 90 Grad von der des ersten Geräusches entfernt liegt. Ist der generierte Wert näher als dieser Mindestabstand am ersten Wert, wird solange eine neue Startposition generiert, bis die Bedingung erfüllt ist.

Panorama Hintergrundbild

Den Hintergrund bildet ein 360-Grad Panoramabild, also eine Rundumsicht einer Landschaft. Der Bildausschnitt der "Fotos" des Spielers beträgt ähnlich wie bei echten Fotografien in etwa 80 Grad. Als Markierung der Rotation dient der linke Rand des Bildausschnittes. Dies bedeutet bei Beginn des Spieles steht der Spieler bei 0 Grad und der linke Rand des Bildausschnittes stimmt mit dem linken Rand des Hintergrundbildes überein. Je nach Position des Spielers beim Beenden der Spielphase, wird auf der Ergebnisseite das Panoramabild so verschoben, dass der linke Rand des Bildaus-

3 Implementierung

schnittes auf der entsprechenden Position liegt. Dabei entspricht logischerweise ein Grad $\frac{1}{360}$ der Panoramabildbreite.

Wie in Abbildung 3.6 zu sehen ist, entsteht ein Problem, falls die Position des Spielers zu weit am rechten Rand des Panoramabildes liegt, etwa zwischen 280 und 360 Grad. Da an diesen Positionen der rechte Bildteil fehlen würde. Um dieses Problem zu lösen wurde auf die einfachste Möglichkeit zurückgegriffen und das sowieso im Cache der Anwendung befindliche Hintergrundbild nochmals rechts neben das Originalbild dupliziert.

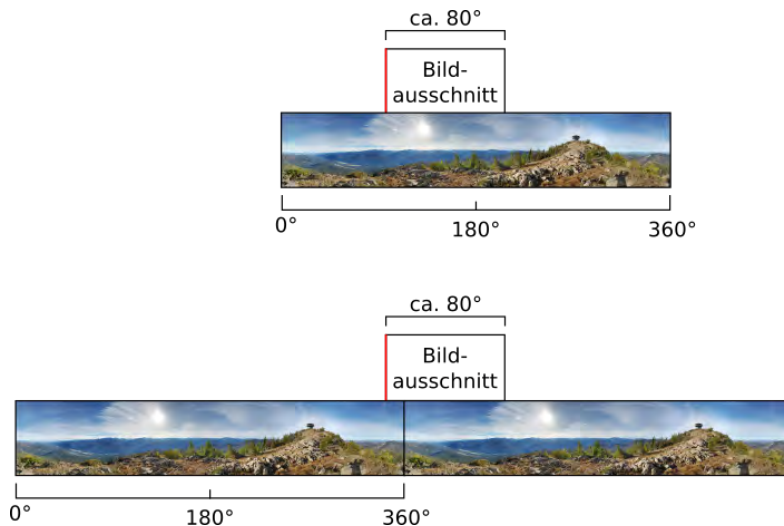


Abbildung 3.6: Sichtbarer Bildausschnitt aus dem Hintergrundbild

3.6 Vergleich zwischen Applikation und Konzept

Nach der eigentlichen Implementierung wird nun zwischen dem ursprünglichen Konzept und der tatsächlich umgesetzten Anwendung verglichen. Dazu werden alle in der finalen Version der Applikation verwendeten Spielelemente einzeln vorgestellt und anschließend die fertigen Seiten nacheinander im Detail beleuchtet.

3.6.1 Spielelemente

Im nachfolgenden Abschnitt werden die einzelnen Elemente gezeigt, die verwendet wurden um die Anwendung dem Konzept entsprechend umzusetzen. Ebenso werden die fertig zusammengesetzten Seiten beleuchtet, die dem Anwender präsentiert werden.

Verwendete Bilder und Geräusche

Wie bereits beschrieben, soll der Benutzer Fotos von zwei verschiedenen Tieren machen können. Damit die Geräusche eindeutig unterscheidbar sind, wurden die Rufe eines Vogels und eines Frosches gewählt. Die gewählten Geräusche besitzen deutlich unterschiedliche Frequenzen und Muster. Die Frequenzbereiche beider Tierrufe sind in Abbildung 3.7 und Abbildung 3.8 zu sehen. Der Vogelruf deckt neben tiefen Tönen auch einem sehr hohen Frequenzbereich ab. Dabei sind relativ kurze Pausen zwischen den Rufen. Das Geräusch des Frosches deckt vor allem tiefe Frequenzen bis in den mittleren Frequenzbereich ab. Dabei erinnert es beinahe an ein Knattern, bei dem die Pausen zwischen den einzelnen Rufen leicht variieren. Der Vogelruf ist der eines Blauhähers (englisch Bluejay), die Gattung des Frosches ist nicht bekannt.

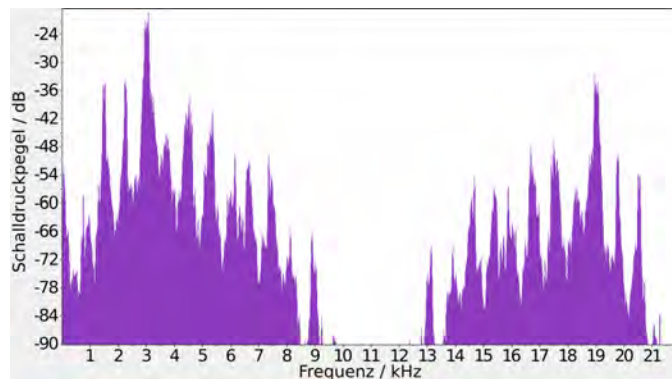


Abbildung 3.7: Frequenzanalyse des Vogelrufes

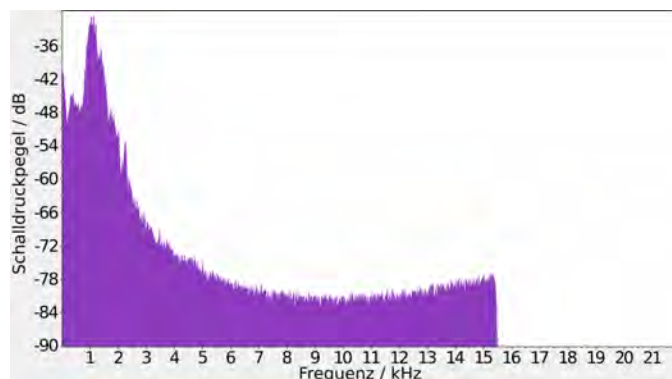


Abbildung 3.8: Frequenzanalyse des Froschrufes

In Abbildung 3.9 und Abbildung 3.10 sind die beiden verwendeten Tierbilder zu sehen. Als zugehöriges Bild für den Vogelruf, wurde ein passendes Bild eines Blauhähers gewählt, für das zweite Geräusch ein entsprechendes Bild eines Frosches. Beide Tiere sitzen auf aus dem unteren Bildrand ragenden Objekten. Dies ist für die Positionierung

3 Implementierung

der Bilder auf der Ergebnis-Seite von Vorteil, da so die Tiere in der Bildmitte abgebildet werden können, ohne dass sie frei im Bild schweben.



Abbildung 3.9: Bild eines Blauhähers für das erste Geräusch



Abbildung 3.10: Bild eines Frosches für das zweite Geräusch

Ein wichtiges Kriterium für das Hintergrundbild, welches auf der Ergebnis-Seite angezeigt wird, ist, dass es sich um ein echtes 360 Grad Panoramabild handelt. So kann der User aus jeder Position ein Foto machen, ohne Wiederholungen im Hintergrundbild zu finden, was die Immersion zerstören würde. Abbildung 3.11 zeigt das verwendete Bild einer Waldkulisse.

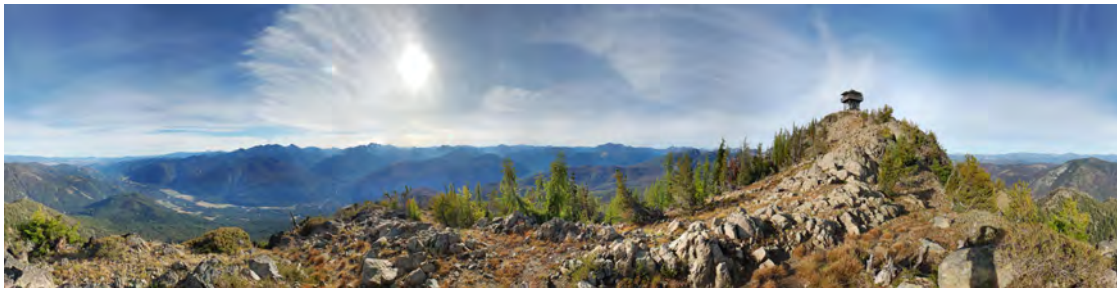


Abbildung 3.11: Verwendetes Panoramabild für den Hintergrund

Die Schaltfläche während der Spielphase ist in Abbildung 3.12 zu sehen. Die verwendete Grafik soll an den Auslöser einer normalen Handykamera erinnern, sodass der Benutzer intuitiv erkennt, dass er ein Foto machen kann.

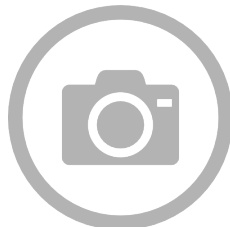


Abbildung 3.12: Piktogramm einer Kamera als Schaltfläche während der Spielphase

Gestaltung der Seiten in der fertigen Anwendung

Um die Logfiles für die Auswertung der Studie anonym einem bestimmten Teilnehmer zuweisen zu können, wird jedem Teilnehmer ein eindeutiger Code zugeteilt. Um diesen für die Logfiles nutzen zu können muss der Benutzercode zu Beginn der Anwendung eingegeben werden. Damit das Hauptmenü nicht überladen wirkt und der Benutzer während des Spielens nicht aus Versehen seinen Code ändern kann, wurde der Applikation eine neue Seite hinzugefügt, die nur beim Start angezeigt wird. Auf dieser Seite, siehe Abbildung 3.13, kann der Benutzer seinen erhaltenen Code eingeben und mit einem Button in das Hauptmenü gelangen. Alle Seiten verwenden die im Styleguide vorgeschriebenen Schriftschnitte und Farben.



Abbildung 3.13: Seite zum Eingeben des Studienrelevanten Benutzer-Codes

Das Hauptmenü, welches in Abbildung 3.14 zu sehen ist, bietet nur die für die Studie relevanten Auswahlmöglichkeiten, welche in einer übersichtlichen Anordnung präsentiert werden. Der Benutzer kann über die beiden Radio-Buttons auswählen ob er ein oder zwei Geräusche suchen möchte. Über eine Checkbox kann der Anwender zusätzlich das Hintergrundgeräusch ein- oder ausschalten, bevor er über die Schaltfläche das Spiel startet.



Abbildung 3.14: Hauptmenü Seite

Die Seite während der Spielphase zeigt, wie in Abbildung 3.15 zu sehen ist, eine Überschrift mit einer Anweisung oberhalb der Auslöser-Schaltfläche. Der Text der Überschrift

3 Implementierung

ändert sich je nach zu suchendem Ziel und gibt dem Spieler somit eine klare Anweisung. Die Schaltfläche zum Auslösen wurde sehr groß platziert, damit der Spieler möglichst schnell auf den Bildschirm tippen kann.



Abbildung 3.15: Bildschirmfoto der Spielphase bei der Suche des ersten Geräusches

Die Ergebnis-Seite zeigt über den gesamten Bildschirm das Ergebnisbild an. Zwei mögliche Ergebnisseiten sind in den Abbildungen 3.16 und 3.17 zu sehen. Um ein echtes Bild einer Kamera zu simulieren, beträgt der Bildausschnitt etwa 80 Grad des Panoramabildes. Auf dem Bildausschnitt wird je nach Ergebnis des Spielers das Tierbild angezeigt. In der unteren linken Ecke sieht der Spieler sein Ergebnis in Zahlen mit der zum Ziel passenden Überschrift. Mit der Schaltfläche kann je nach Einstellung das Spiel fortgesetzt ("Weiter suchen") oder zum Hauptmenü zurückgekehrt ("Nochmal") werden.

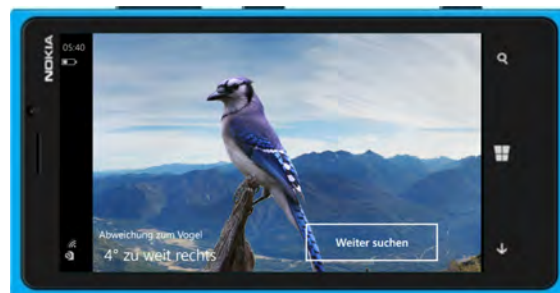


Abbildung 3.16: Ergebnis-Seite bei der Suche des ersten Geräusches

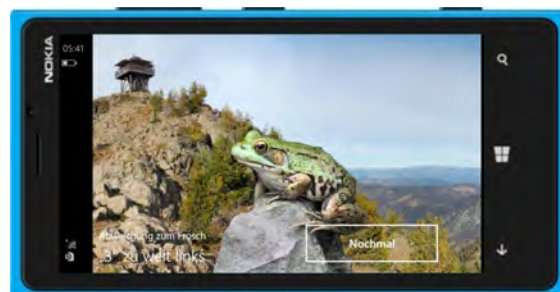


Abbildung 3.17: Ergebnis-Seite bei der Suche nach dem zweiten Geräusch

3.7 Dokumentation für die Wiederverwendbarkeit

Für eine eventuelle Wiederverwendung des entwickelten Prototyps, werden im Folgenden die wichtigsten Punkte zum Software Development erläutert.

3.7.1 Softwarekonfigurationen

Das Programm wurde mit "Visual Studio 2010 Express for Windows Phone" entwickelt. Es wurden ebenso die Versionen Visual Studio 2013 Community und Visual Studio 2015 Community getestet und als kompatibel befunden. Die Anwendung wurde mit dem .NET-Framework 4.5 und dem Microsoft.XNA-Framework realisiert. Für die Entwicklung wurde das Windows Phone SDK 8.0 verwendet, Windows Phone SDK 8.1 ist mit der Anwendung ebenfalls kompatibel. Als Betriebssystem für das Windows Phone wurden sowohl Version 8.0, als auch 8.1 getestet, wobei keinerlei Unterschiede festzustellen waren.

Um die Logfiles auszulesen, wurde das Programm Windows Phone Power Tools 2.83 verwendet.

3.7.2 Die Anwendung auf einem Windows Phone installieren

Falls das Windows Phone für App-Entwicklung registriert ist, lässt sich die Anwendung am schnellsten direkt über Visual Studio auf dem Smartphone installieren. Hierzu muss das Smartphone zunächst über ein USB-Kabel mit dem PC verbunden werden. Anschließend kann die Anwendung über einen Klick auf die Schaltfläche mit dem grünen Dreieck, wie er in Abbildung 3.18 zu sehen ist, direkt auf dem Windows Phone installiert werden. Dieser Bereich befindet sich standardmäßig in der oberen Menüleiste von Visual Studio. Dabei ist sicherzustellen, dass bei der Art des Exports, im ersten Dropdown-Menü "Release" und als Ort "Device" auf dem Button ausgewählt ist. Alternativ befindet sich eine "<Projektname> erstellen"-Schaltfläche im Reiter "Erstellen" in der oberen Menüleiste von Visual Studio.

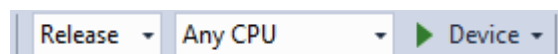


Abbildung 3.18: Menüleiste zum Exportieren der Anwendung in Visual Studio

Ist das Windows Phone nicht für App-Entwicklung registriert, so lässt sich die Anwendung mittels einer SD-Karte auf dem Gerät installieren. Hierzu muss die Anwendung

3 Implementierung

zuerst wie oben erstellt werden. Dies generiert eine XAP-Datei, welche im Projektordner des Visual Studio Projektes der Anwendung im Unterordner Bin/Release zu finden ist. Nun kopiert man die XAP-Datei auf eine SD-Karte und legt diese anschließend in das Smartphone ein. Alternativ kann die Datei auch auf den internen Gerätespeicher kopiert werden. Anschließend öffnet man die App "Store" auf dem Gerät und wählt unter Weitere Optionen (drei Punkte am unteren rechten Bildschirmrand) die Option "Lokale Apps installieren". Wählt man die entsprechende App aus, wird diese auf dem Gerät installiert [16].

3.7.3 Logfiles

Die Logfiles werden nach der ID benannt, die der Studienteilnehmer zu Beginn mit dem Ausfüllen der Einverständniserklärung erhalten hat. Sie werden in das Wurzelverzeichnis des "Isolated Storage" der Windows Anwendung gespeichert. Der "Isolated Storage" ist Teil des .NET-Frameworks und bezeichnet einen Bereich der Festplatte, welcher nur für eine Anwendung verfügbar ist [29].

Die Logfiles werden zur Laufzeit der Anwendung dynamisch erweitert, sobald Daten erfasst werden. Aufgebaut sind die Logfiles nach einem CSV-Schema mit den folgenden Werten in dieser Reihenfolge.

ID: Eine einmalige Ganzzahl pro Teilnehmer

Zeitstempel: Zeit im Format dd:MM:YYYY hh:mm:ss zu welcher der Benutzer den Button während der Spielphase gedrückt hat

Fall-ID: Dieser Wert speichert die Einstellungen, die im Hauptmenü ausgewählt wurden. Die Fall-ID kann die Werte 0, 1, 2 und -1 annehmen. Dabei bedeutet

- 0 ein Ziel (Vogel), kein Hintergrundgeräusch
- 1 zwei Ziele, kein Hintergrundgeräusch
- 2 zwei Ziele mit Hintergrundgeräusch
- 1 andere Kombination, die nicht in der Studie abgefragt wurde

Ziel-ID: Beschreibt das Geräusch, dass der Benutzer als Ziel hatte. Ist der Wert 0, wurde der Vogel gesucht. Ist der Wert 1, so war das Froschgeräusch das Ziel

Zeit: Zeit in Millisekunden die benötigt wurde um das gesuchte Geräusch zu lokalisieren und auf den Button auf der Spiel Seite zu drücken

Abweichung: Dieser Wert zeigt die Differenz in Grad zwischen der wirklichen Position des Geräusches und dem Ergebnis des Benutzers. Eine positive Zahl bedeutet,

dass sich der Benutzer zu weit rechts befand, eine negative Zahl hingegen, dass er zu weit links positioniert war. Diese Abweichung wird mit einer Nachkommastelle gespeichert

Beispiel eines Logfiles mit dem Namen 21.txt

```
1 21; 29.06.2015 16:34:11; 0; 0; 10852; 1,9
2 21; 29.06.2015 16:34:25; 0; 0; 7283; 8,2
3 21; 29.06.2015 16:34:34; 0; 0; 6726; -0,9
4 21; 29.06.2015 16:34:52; 1; 0; 8688; -8,9
5 21; 29.06.2015 16:35:03; 1; 1; 9511; -0,5
6 21; 29.06.2015 17:16:47; 1; 0; 6856; -9,2
7 21; 29.06.2015 17:16:58; 1; 1; 9865; -1,8
8 21; 29.06.2015 17:17:10; 1; 0; 8285; 12,6
9 21; 29.06.2015 17:17:21; 1; 1; 10530; -4,2
10 21; 29.06.2015 17:17:32; 2; 0; 8259; -10,2
11 21; 29.06.2015 17:17:40; 2; 1; 6486; -12,6
12 21; 29.06.2015 17:17:49; 2; 0; 5859; 17,8
13 21; 29.06.2015 17:17:58; 2; 1; 7601; -46,5
14 21; 29.06.2015 17:18:11; 2; 0; 8343; 2,3
15 21; 29.06.2015 17:18:22; 2; 1; 9899; 4,1
```

Code 3.4: Inhalt eines Logfiles

3.7.4 Codestruktur

Der Quellcode der Anwendung besteht aus sechs Klassen. Vier dieser Klassen beinhalten den jeweiligen Code der verwendeten Seiten. Die beiden verbleibenden Klassen sind Hilfsklassen. Die Zusammenhänge zwischen den Klassen, welche in Abbildung 3.19 auf Seite 40 grafisch dargestellt sind, werden im Folgenden erklärt.

Eine der Hilfsklassen ist der sogenannte Logger. In dieser Klasse werden alle für die Logfiles benötigten Daten zusammengetragen und zur Laufzeit in diese geschrieben. Da die benötigten Daten aus fast allen verschiedenen Seiten-Klassen stammen, besitzt die Logger-Klasse mehrere statische Methoden, die von jeder anderen Klasse der Anwendung aufgerufen werden können.

Die Klasse "Soundquelle" beinhaltet einen Konstruktor für das Objekt "Soundquelle", welches, wie bereits in Code 3.3 auf Seite 29 erwähnt, folgende Werte beinhaltet:

3 Implementierung

- `string` name
- `float` position
- `SoundEffect` sound
- `Image` image

Dabei gibt der String "name" die Bezeichnung des zu suchenden Tieres an, zum Beispiel "Vogel". Der float "position" enthält die bereits in Abschnitt 3.5.2 beschriebene Startposition des Objektes. Dem Konstruktor werden außerdem die URIs des zu verwendeten Bildes und der Sounddatei übergeben. Im Anschluss werden entsprechende Instanzen der Image- bzw. SoundEffect-Klasse intern erstellt. Die SoundEffect-Klasse wird für die Wiedergabe von räumlichen Geräuschen benötigt.

Beim Starten der Anwendung gelangt der Spieler zuerst auf die Seite der Code Eingabe. In der zugehörigen Klasse wird lediglich der Code an den Logger weitergegeben, der den Wert intern speichert und daraufhin ein nach dem Code benanntes Logfile erstellt. Anschließend navigiert die Code Eingabe Klasse direkt zum Hauptmenü.

Die im Hauptmenü gewählten Einstellungen, ob ein oder zwei Geräusche mit oder ohne Hintergrundgeräusch gesucht werden sollen, werden in zwei Booleans gespeichert. Der Boolean "oneSound" wird true gesetzt, falls nur ein Geräusch gesucht werden soll und entsprechend false, bei zwei Geräuschen. Der Boolean "bgSound" wird auf true gesetzt, falls ein Hintergrundgeräusch abgespielt werden soll.

Diese beiden Werte werden bei Navigation auf die Spielseite mitgegeben. Dies funktioniert mit Hilfe des PhoneApplicationService, einer im Microsoft.Phone.Shell Namespace enthaltenen Hilfsklasse.

Mit der Property "PhoneApplicationService.Current.State["Bezeichner"]" ist es möglich, ähnlich wie bei einem HTML-POST, ein beliebiges Objekt an eine andere Seite weiterzugeben.

In der Klasse der Spielseite, wird zunächst die Combined Motion API gestartet, mit deren Hilfe die Rotationsänderungen des Smartphones erkannt werden. Anschließend werden je nach getroffener Einstellung, bzw. übergebenem Boolean, ein oder zwei neue Instanzen der Klasse "Soundquelle" erstellt. Darauf hin werden der angezeigte Text an die Aufgabe angepasst, die SoundEmitter und der SoundListener, wie in Abschnitt 3.5.2 beschrieben positioniert, die entsprechenden Geräusche abgespielt und die für die Logfiles wichtige Stoppuhr gestartet.

Nun wird bei jeder Veränderung der Lage des Smartphones der bzw. die SoundEmitter entsprechend bewegt und anschließend getestet, ob sie hinter dem Spieler liegen um eventuell die Lautstärke des Geräusches zu reduzieren.

Beim Drücken des Buttons, wird zunächst die Stoppuhr angehalten und der benötigte Speicher der Motion API freigegeben. Beim Navigieren auf die Ergebnisseite werden nachfolgende Werte mitgegeben.

Der Winkel, in welchem sich der Spieler beim Drücken des Buttons befand, wird im Float "angle" gespeichert. Die gemessene Zeit der Stoppuhr wird als String in der Variable "timeNeeded" weitergegeben, ebenso wie die Soundquelle "sound1", welche die erste zu suchende Geräuschquelle ist.

Waren zwei Geräusche aktiv, wird außerdem zusätzlich noch die zweite Soundquelle namens "sound2" übermittelt, zusammen mit einem Boolean "firstTime", welcher, sofern auf true gesetzt, angibt, dass das erste Geräusch gesucht wurde.

In der Klasse der Ergebnisseite, wird zunächst der Abstand in Grad zwischen der Position des gesuchten Geräusches und der übermittelten Position des Benutzers berechnet, auf eine Nachkommastelle gerundet und an die Logger-Klasse übergeben. Anschließend werden die angezeigten Texte geändert, sodass sie die Abweichung und die Richtung zum gesuchten Tier angeben.

Nachdem das Hintergrundbild geladen wurde, wird dessen Position relativ zur Drehung des Benutzers berechnet und dem zur Anzeige verwendeten Canvas, der die Größe des gesamten Displays hat, hinzugefügt. Im Vordergrund werden danach die entsprechenden Bilder der zu suchenden Tiere an den korrekten Positionen dem Canvas hinzugefügt.

Durch Tippen des Buttons navigiert die Klasse abhängig vom übermittelten Boolean "firstTime". Hat dieser den Wert true, bedeutet dies, dass es sich um den ersten Durchgang von zwei zu suchenden Geräuschen handelt. Dementsprechend navigiert die Klasse zurück zur Spielseite und übergibt die beiden Soundquellen "sound1" und "sound2" sowie einen Boolean "secondTime" = true. Dort werden anschließend keine neuen Soundquellen initiiert und der Benutzer muss die zweite Soundquelle suchen.

Ist der Boolean "firstTime" false, so navigiert die Ergebnis-Klasse zum Hauptmenü. Von dort kann anschließend ein neues Spiel gestartet werden.

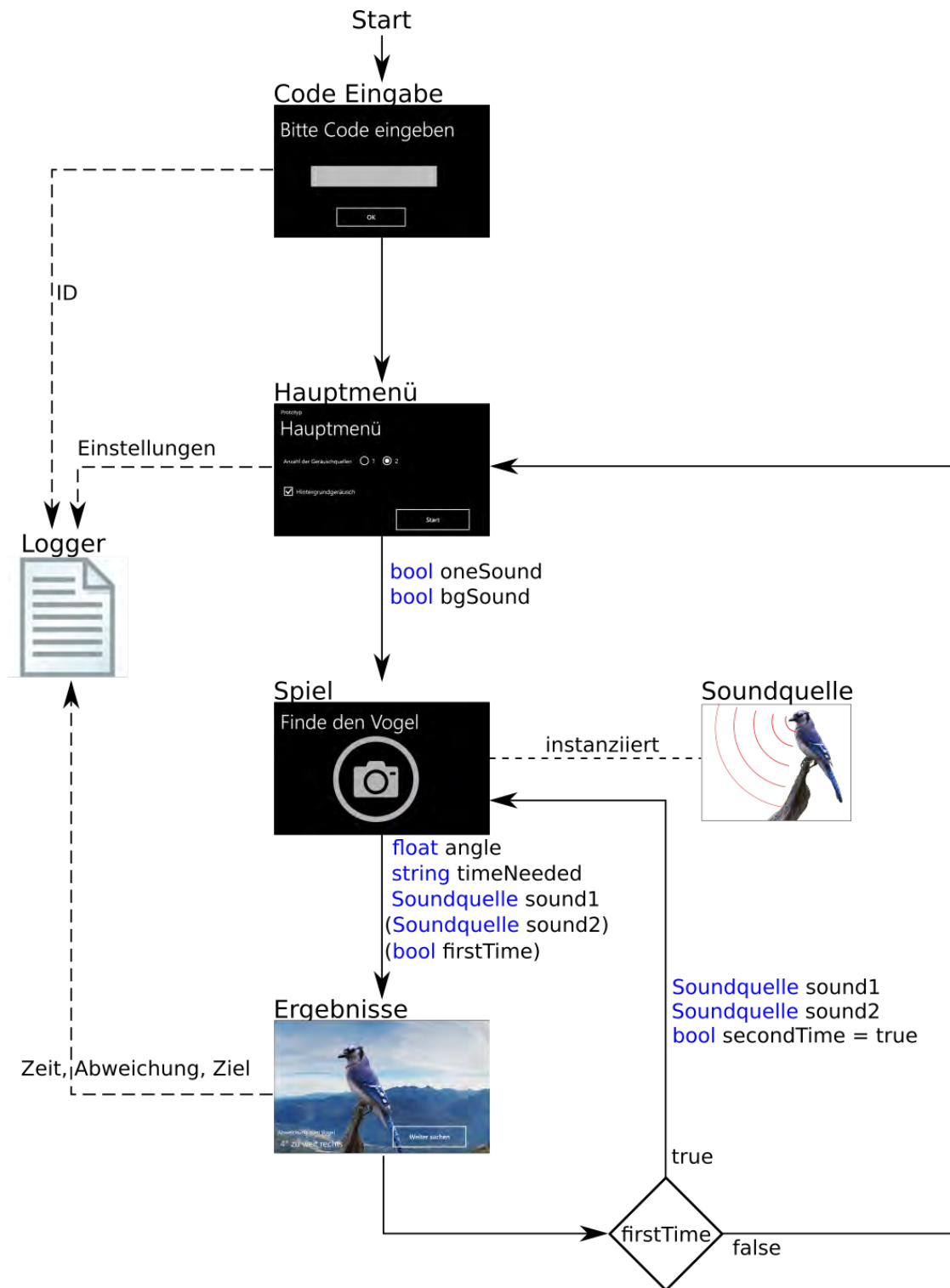


Abbildung 3.19: Schematischer Aufbau des Quellcodes

3.7.5 Bekannte Probleme

Bei Tests der Anwendung wurde festgestellt, dass die Rotationserkennung nicht optimal funktioniert wenn das Gerät komplett vertikal gehalten wird. Wesentlich bessere Ergebnisse erhält man, wenn man das Gerät, während man es im Querformat hält, etwas nach hinten kippt. Man sollte also mit der Rückseite des Gerätes leicht Richtung Boden deuten.

Ebenso wirken sich schnelle ruckartige Bewegungen, beispielsweise eine schnelle Drehung um 180 Grad, negativ auf die Präzision der Rotationserkennung aus, wie bei Testläufen festgestellt wurde. Für zu schnelle Bewegungen sind die im Gerät verbauten Sensoren zu träge.

4 Studie

Das Konzept der Anwendung wurde im Rahmen weiterer Abschlussarbeiten auf den Plattformen Android, iOS und HTML5 ebenfalls umgesetzt. Um einen Vergleich der verschiedenen Implementierungen und eine Untersuchung der Funktion der Anwendungen zu erlangen, wurde eine begleitende Studie durchgeführt. In diesem Kapitel werden nun zunächst der Aufbau und Ablauf der Studie und anschließend die Ergebnisse präsentiert.

In der Studie soll neben dem Vergleich der vier Implementierungen auch untersucht werden, ob die Anzahl, sowie die Auswahl der zu suchenden Geräusche einen Einfluss auf die Geschwindigkeit und die Präzision der Versuchsteilnehmer hat.

4.1 Aufbau

Die Versuchspersonen mussten zunächst in Form einer einen Fragebogen ausfüllen und eine Einverständniserklärung über die anonyme Nutzung ihrer erhobenen Daten unterzeichnen. Auf Grund des geringen Umfangs des Fragebogens und der Verwendung der angegebenen Daten für alle Plattformen, wurde eine Paper-Pencil-Lösung einer in der Anwendung implementierten Befragung vorgezogen [26]. Auf dem Fragebogen wurde den Teilnehmern eine Identifikationsnummer zugewiesen, die sie beim Start der zu testenden Anwendungen einzutragen hatten. Die Teilnehmer wurden zunächst nach soziodemographischen Daten, wie Alter und Geschlecht gefragt. Weiter wurde nach eventuellen Hörerkrankungen, der Erfahrung mit Mobilgeräten, der privat benutzten mobilen Plattform und der Erfahrung mit Videospiele gefragt. Der gesamte Fragebogen befindet sich im Anhang A.1 der Arbeit. Um technische Störungen zu minimieren, wurde der Versuch im Freien durchgeführt, da der geräteinterne Kompass in geschlossenen Räumen schlechter reagiert. Während den Durchläufen der Versuchspersonen, werden die in Abschnitt 3.7.3 beschriebenen Logfiles angelegt. Diese liefern exakte Werte für die Auswertung und es kann auf eine weitere Paper-Pencil-Befragung verzichtet werden [25].

4.2 Ablauf

Die Studienteilnehmer müssen in randomisierter Reihenfolge auf jeder der vier Plattformen die gleichen Aufgaben erfüllen. Da sich die Teilnehmer bei ihrem ersten Versuch noch an das Bedienungskonzept gewöhnen müssen, ist davon auszugehen, dass diese Durchgänge etwas schlechtere Werte liefern. Mit der zufälligen Reihenfolge wird jede der Plattformen etwa gleich häufig als erste Plattform von einem Teilnehmer getestet. Somit relativieren sich eventuelle Anfangsprobleme.

Die Versuchspersonen mussten je drei mal

- ein Geräusch (Vogel), ohne Hintergrundgeräusch
- zwei Geräusche (erst Vogel, dann Frosch), ohne Hintergrundgeräusch
- zwei Geräusche, mit Hintergrundgeräusch

suchen. Demnach liefert jeder Teilnehmer 15 Datensätze mit Abweichung und benötigter Zeit auf jeder der vier Plattformen.

4.3 Hypothesen

Für die Hypothesen, sowie die Berechnungen werden die Begriffe "Fall" und "Ziel", bzw. "Fall-ID" und "Ziel-ID", wie durch die Logfiles definiert verwendet (vgl. Abschnitt 3.7.3). Die aufgestellten Hypothesen der Studie lauten wie folgt.

Hypothese 1 Die benötigte Zeit ist signifikant abhängig vom gewählten Fall.

Hypothese 2 Die benötigte Zeit ist signifikant abhängig vom gesuchten Ziel.

Hypothese 3 Die Winkeldifferenz ist signifikant abhängig vom gewählten Fall.

Hypothese 4 Die Winkeldifferenz ist signifikant abhängig vom gesuchten Ziel.

4.4 Ergebnisse

Zur Untersuchung der Hypothesen wurden Varianzanalysen mit den auf dem Windows Phone gesammelten Daten durchgeführt. Dabei wurde jeweils eine Varianzanalyse oder ANOVA (vom englischen analysis of variance) für die Bearbeitungszeit und die Winkelabweichung gerechnet. Für die Auswertung wird ein Signifikanzniveau von 5%

festgelegt, unter welchem das Ergebnis als signifikant betrachtet und die Hypothese nicht abgelehnt wird.

4.4.1 Bearbeitungszeiten

Tabelle 4.1 zeigt die Varianzanalyse zur Untersuchung des Einflusses von Fall und Ziel auf die benötigte Zeit der Versuchspersonen. Die Festen Effekte sind dabei Fall-ID, bzw. Ziel-ID. Den einzigen zufälligen Effekt stellt die Versuchsperson dar.

	numDF	denDF	F-Wert	p-Wert
(Intercept)	1	346	115.27904	<.0001
Fall-ID	2	346	0.56422	0.5693
Ziel-ID	1	346	6.65582	0.0103

Tabelle 4.1: Varianzanalyse der benötigten Zeit

Mit einer Irrtumswahrscheinlichkeit von ca. 1%, hat das gesuchte Ziel einen signifikanten Einfluss auf die benötigte Zeit. In Tabelle 4.2 sind die gemittelten Zeiten aller Versuchspersonen des Windows Phones aufgelistet, die benötigt wurden um das jeweilige Ziel zu finden. Durch den Studienablauf bedingt, liegen für das erste Ziel (Vogel) mehr Messwerte vor.

Ziel	Zeit
Vogel	15,1
Frosch	17,4

Tabelle 4.2: Gemittelte benötigte Zeit zum suchen der Ziele

Um die Signifikanz zu prüfen, wird ein zweiseitiger T-Test für zwei Stichproben mit gleicher Varianz durchgeführt. Der T-Test liefert einen p-Wert von 0,098 und liegt demnach über dem festgelegten Signifikanzniveau.

4.4.2 Winkeldifferenzen

In Tabelle 4.3 wird die Varianzanalyse zur Untersuchung des Einflusses von Fall und Ziel auf die Winkeldifferenz gezeigt. Für die Berechnungen der Winkeldifferenzen wurden zunächst die absolute Winkeldifferenz gebildet und somit die Vorzeichen ignoriert. Dies

4 Studie

verhindert Fehler bei der Mittelwertbildung. Die Festen Effekte sind durch Fall-ID, bzw. Ziel-ID gegeben. Den einzigen zufälligen Effekt stellt wiederum die Versuchsperson dar.

	numDF	denDF	F-Wert	p-Wert
(Intercept)	1	346	72.03980	<.0001
Fall-ID	2	346	3.00570	0.0508
Ziel-ID	1	346	4.19458	0.0413

Tabelle 4.3: Varianzanalyse der absoluten Winkeldifferenz

Der p-Wert des Einflusses des Ziels auf die Winkeldifferenz liegt unterhalb des festgelegten Signifikanzniveaus von 5%. Um die Signifikanz zu prüfen wird ebenfalls ein zweiseitiger T-Test durchgeführt. Mit einem p-Wert von 0,0087 ist der Einfluss des zu suchenden Geräusches auf die Präzision hochsignifikant.

Tabelle 4.4 zeigt die Mittelwerte der absoluten Winkeldifferenz aller Studienteilnehmer auf der mobilen Windowsplattform für das jeweilige Ziel. In Abbildung 4.1 auf Seite 47 sind die Unterschiede in Zeit und Winkeldifferenz zwischen den beiden zu suchenden Geräuschen grafisch dargestellt. Dabei ist klar die große Differenz der Präzision bei den zwei Zielen zu erkennen.

Ziel	Winkeldifferenz
Vogel	18,7
Frosch	26,5

Tabelle 4.4: Gemittelte absolute Winkeldifferenz der Ziele

4.4.3 Vergleich aller Betriebssysteme

Wie bereits erwähnt, war die Studie auch dazu ausgelegt, die verschiedenen Implementierungen der Anwendung zu testen und zu vergleichen. Für den Vergleich wurden die Logfiles aller Plattformen zusammengefügt. Um den Durchschnitt einzelner Plattformen nicht zu verfälschen, wurden überzählige Messwerte aus der Tabelle entfernt, sodass alle Plattformen etwa die gleiche Anzahl an Datensätzen haben. Anschließend wurden die Mittelwerte über alle Studienteilnehmer für die benötigte Zeit und die Winkeldifferenz gebildet. Diese werden in Tabelle 4.5 für jede Plattform aufgelistet. In Abbildung 4.2 auf Seite 48 werden diese Daten gegeneinander aufgetragen dargestellt.

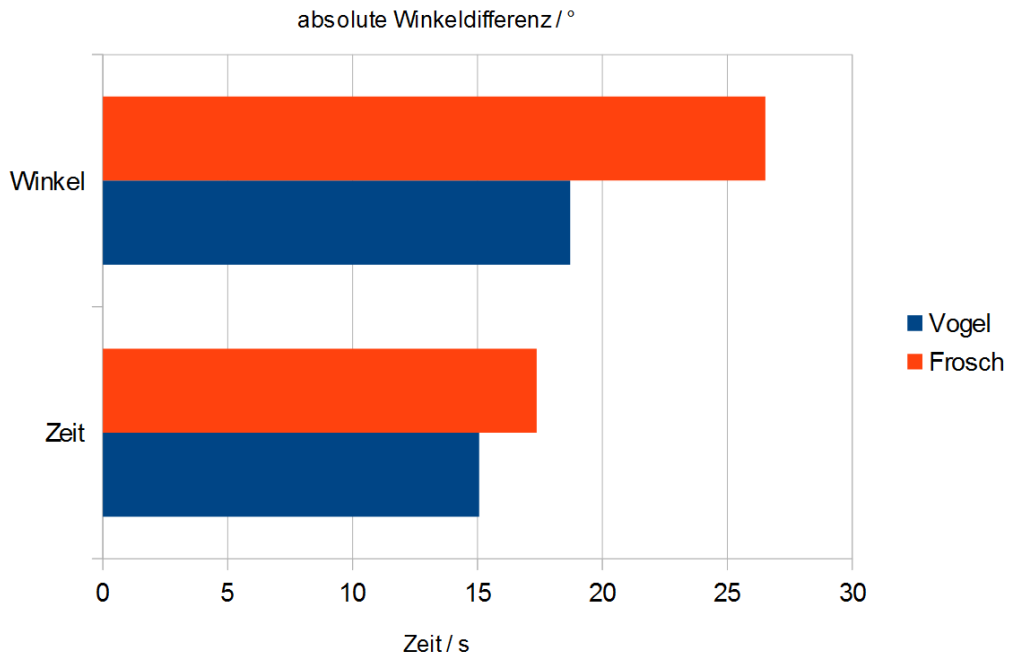


Abbildung 4.1: Zeit- und Winkeldifferenzunterschiede zwischen zu suchenden Geräuschen

Betriebssystem	benötigte Zeit	Winkeldifferenz
Android	12.12	19.6
iOS	9.74	13.3
Web	14.50	55.5
Windows	15.97	21.8

Tabelle 4.5: Bearbeitungszeiten und Winkeldifferenzen der einzelnen Betriebssysteme im Vergleich

4.5 Diskussion der Ergebnisse

Nach der Vorstellung der Studienergebnisse werden diese im nachfolgenden Abschnitt in Hinsicht auf die Durchführung und die aufgestellten Hypothesen besprochen. Im Anschluss wird ein Vergleich zwischen den verschiedenen Plattformen gezogen.

4.5.1 Schwächen der Studie

Die Studie wies in ihrer Vorbereitung und Durchführung einige Probleme auf.

- Wie bereits in Abschnitt 3.7.5 genannt, reagiert die Rotationserkennung unpräzise, falls das Gerät zu gerade gehalten und zu ruckartig bewegt wird. Zwar wurden die

4 Studie

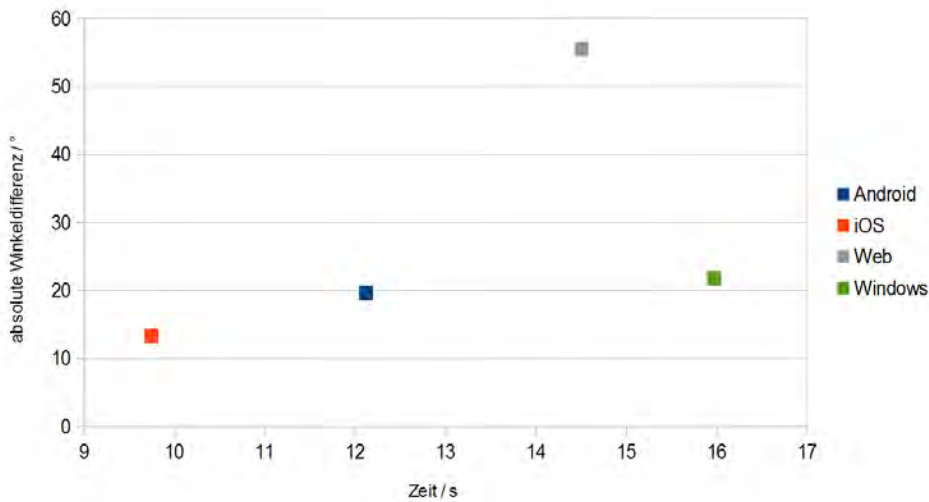


Abbildung 4.2: Zeit- und Winkeldifferenzunterschiede zwischen den Betriebssystemen

Versuchspersonen darauf hingewiesen, jedoch beachteten einige Teilnehmer die Ratschläge nicht, was zu ungenauen Ergebnissen führt.

- Die Studie dauerte durch die benötigte Anzahl der Messungen auf allen Geräten von jedem Teilnehmer recht lange. Somit ist es möglich, dass die Konzentration der Studienteilnehmer besonders bei den letzten Durchgängen nachließ und die Ergebnisse somit ungenauer wurden.
- Die Implementierungen für HTML5 und iOS unterscheiden sich zum Teil stark vom vereinbarten Konzept. So endet beim Suchen von zwei Geräuschen die Wiedergabe des Vogelrufs, sobald dieser gefunden wurde. Im Gegensatz zu den anderen Umsetzungen ist hier bei der Suche nach dem Frosch nur noch dieser zu hören, jedoch nicht mehr der Vogel. Dies erleichtert die Ortung des zweiten Geräusches auf jenen Plattformen.

4.5.2 Betrachtung der Hypothesen

Die ersten drei Hypothesen konnten nicht gestützt werden. Die Varianzanalyse bezüglich der Bearbeitungszeit aus Tabelle 4.1 in Abschnitt 4.4.1 zeigt, dass die Signifikanz der Fall-ID über dem festgelegten Signifikanzniveau liegt. Damit kann Hypothese 1 nicht bestätigt werden. Der durchgeführte T-Test gibt zu erkennen, dass die Ziel-ID ebenfalls keinen signifikanten Einfluss auf die benötigte Zeit ausübt. Somit ist Hypothese 2 ebenfalls nicht verifizierbar. Die winkelbezogene Varianzanalyse (Tabelle 4.3) in Abschnitt 4.4.2 zeigt auf, dass der Fall, also die gewählte Einstellung, wie viele Geräusche

gesucht werden müssen, ebenfalls keinen signifikanten Einfluss auf die Winkeldifferenz ausübt. Dies widerlegt Hypothese 3.

Die vierte Hypothese konnte jedoch gestützt werden. So war ein hochsignifikanter Einfluss des zu suchenden Geräusches auf die Präzision der Studienteilnehmer festzustellen. Dies bedeutet, dass das für den Frosch verwendete Geräusch schwerer zu orten war, als das verwendete Vogelgeräusch.

Allgemein bedeuten diese Ergebnisse, dass es für die Ortung eines bestimmten Geräusches nicht wichtig ist, ob sich zeitgleich ein weiteres Geräusch im Raum um den Benutzer bewegt. Die bestätigte Hypothese 4 zeigt, dass das gewählte Geräusch einen großen Einfluss auf die Präzision des Anwenders hat. In zukünftigen Versionen ähnlicher Anwendungen, sollte ein vermehrtes Augenmerk auf die Auswahl der Geräusche gelegt werden.

4.5.3 Vergleich zwischen den Plattformen

Im Gegensatz zu der in dieser Arbeit beschriebene Implementierungsform ausschließlich mit den im .NET- und Microsoft.xna-Framework zur Verfügung gestellten Klassen und Methoden, wurden die anderen Anwendungen mit Hilfe der OpenAL API implementiert. Diese ist eine offene Schnittstelle zu Audio Hardware, die viele Methoden für die Wiedergabe von räumlichen Geräuschen liefert [8] [22]. Im Vergleich zwischen allen Plattformen, wie in Abbildung 4.2 aufgetragen, fällt auf, dass sich die Winkeldifferenz von Android, iOS und Windows Phone nicht sehr unterscheiden, wobei iOS eindeutig die besten Ergebnisse lieferte. Die Web-Implementierung mittels HTML5 ist im direkten Vergleich weit abgeschlagen, jedoch liegen die Ursachen hierfür in den begrenzten Möglichkeiten der Webschnittstelle für mobile Anwendungen.

Im Vergleich der benötigten Zeiten fällt wiederum auf, dass die iOS-Implementierung die geringste mittlere Bearbeitungszeit aller Studienteilnehmer benötigte. Die Android Anwendung liegt nur wenige Sekunden darüber. Einen deutlich längere Zeit benötigten die Benutzer auf der Web- und der Windows-Applikation.

Ein Erklärungsversuch dieser langen benötigten Zeiten auf dem Windows Phone könnten die zur Studiendurchführung verwendeten Kopfhörer sein. Jede Plattform benutzte unterschiedliche Kopfhörer zur Ortung der Geräusche. So waren die beim Windows Phone benutzten Kopfhörer im Vergleich zu denen der anderen Geräte recht leise, auch bei maximaler Lautstärke des Smartphones. Dies wurde während der Durchführung

4 Studie

ebenfalls von einigen Studienteilnehmern angesprochen. Die im Vergleich geringere Lautstärke könnte eine Ursache der längeren benötigten Suchzeit sein, da sich die Teilnehmer stärker auf die Geräusche konzentrieren mussten.

5 Zusammenfassung

In diesem Kapitel werden die einzelnen Abschnitte des Entwicklungs- und Evaluierungsprozesses rekapituliert.

5.1 Implementierung

Die Entwicklung für Windows Phone setzt die Benutzung des .NET-Frameworks voraus. Dieses und das Microsoft.XNA-Framework bieten mit dem Windows Phone SDK alle notwendigen Funktionen um jede Art von Anwendung für Windows Phone zu entwickeln.

Jedoch sind die angebotenen Funktionen nicht sehr transparent. Die vorhandene Dokumentation listet zwar alle Klassen und zugehörige Funktionen auf, jedoch werden die im Hintergrund ablaufenden Prozesse nicht beleuchtet.

Das bedeutet, der Programmierer kann zwar recht schnell funktionsfähige Anwendungen mit Hilfe der angebotenen Schnittstellen erstellen, weiß jedoch nicht ganz eindeutig, welche Berechnungen im Hintergrund ablaufen.

In Hinsicht auf die Studienergebnisse, waren die Implementierungen, die OpenAL als API für die Realisierung der 3D-Geräusche verwendet haben, etwas präziser als die in der Windows Phone verwendeten Variante.

5.1.1 Realisierung der Anwendung

Zunächst wurden die Kernfunktionen der Anwendung, besonders die Rotationserkennung mit unmittelbarem optischen Feedback, durch vertikales Prototyping getestet. Im Anschluss wurde ein evolutionärer Ansatz genutzt, um mit Hilfe der Erkenntnisse aus dem Prototyp, eine lauffähige Version mit allen vorher definierten Features zu implementieren.

5.1.2 Anforderungsabgleich

In diesem Abschnitt wird betrachtet, ob die in der Anforderungsanalyse in Abschnitt 3.2 genannten Features in der finalen Fassung der Anwendung implementiert wurden. Dazu stellt Tabelle 5.1 in einer erweiterten Form von Tabelle 3.1 eine Übersicht bereit.

Feature	muss	soll	kann	Implementiert
Räumliche Positionierung von Tönen	x			ja
Leiser, wenn Ton hinten		x		ja
Auswahl ob ein oder zwei Geräusche	x			ja
Mehrere auswählbare Geräusche			x	nein
Hintergrundbild wählbar			x	nein
Filter auf Töne anwenden			x	nein
Nur Touchscreen-Steuerung			x	nein
Visuelles Feedback	x			ja
Anzeige der Abweichung in Grad		x		ja
Logfiles anlegen	x			ja

Tabelle 5.1: Muss-, Soll-, Kann-Klassifikation der Features mit Implementierungsstatus

Bewertung

In der finalen Version wurden alle mit "muss" gekennzeichneten Features umgesetzt. Ebenso wurden alle mit "soll" klassifizierten Features implementiert, da sie der Benutzererfahrung zuträglich sind. Die nicht implementierten Funktionen waren für die begleitende Studie, zu deren Zweck die Anwendung entwickelt wurde, nicht relevant. Sie hätten nur zusätzliche Optionen geboten, welche die Studienteilnehmer eventuell bei der Durchführung der Studie abgelenkt hätten.

Dennoch können die nicht implementierten Features als mögliche Ergänzungen einer zukünftigen Version betrachtet werden, sollte das Spielkonzept für einen anderen Verwendungszweck wieder aufgegriffen werden.

5.1.3 Tools

Rückblickend betrachtet war Visual Studio 2012 Express als Hauptentwicklungs-Tool eine gute Wahl. Dank der Trennung von Benutzeroberfläche im XAML (Extensible Appli-

cation Markup Language) Schema und Quellcode, war es möglich sehr früh im Entwicklungsprozess bereits Prototypen zu erstellen. Visual Studio besitzt einen sogenannten XAML Designer. Ein Graphisches Tool, mit dem die Benutzeroberfläche via Drag & Drop schnell erstellt und angepasst werden kann. Im nebenstehenden XML basierten Code werden die Änderungen in Echtzeit übernommen. Ebenso sieht man die getätigten Änderungen im Code sofort auf der Designer-Seite. Die Anwendung kann direkt aus Visual Studio heraus auf dem Smartphone getestet werden, was ein sehr schnelles und unmittelbares Debuggen ermöglicht.

Zum Auslesen der Daten aus dem isolierten Speicher der Anwendung wurden sowohl der Windows Isolated Storage Explorer, als auch die Windows Phone Power Tools getestet. der Isolated Storage Explorer ist ein Kommandozeilen Tool, welches mit dem Windows Phone SDK installiert wird [15]. Zwar können mit dem Tool alle nötigen Befehle ausgeführt werden, dennoch ist die Bedienung des Power Tools dank seiner grafischen Oberfläche übersichtlicher und intuitiver.

6 Fazit und Ausblick

Zum Abschluss der Arbeit wird nachfolgend zunächst ein Fazit aus den Studienergebnissen gezogen. Im Anschluss wird ein Ausblick auf die zukünftigen Möglichkeiten gewährt, welche sich mit einer Weiterentwicklung der in dieser Arbeit konzipierten mobilen Anwendung eröffnen.

6.1 Fazit aus der Studie

Aus dem Vergleich aller Plattformen kann man zusammenfassend sagen, dass das angedachte Konzept grundlegend funktioniert. Die Zuverlässigkeit einer präzisen Ortung der Geräusche ist zum aktuellen Stand aller Anwendungen noch nicht gegeben, da selbst die iOS-Implementierung, welche die besten Ergebnisse aufwies, eine gemittelte Abweichung von ca. 13 Grad aufweist. Jedoch ist der grundlegende Zweck der Anwendung die auditorische Stimulation des Users, sodass sich dieser auf eine externe Geräuschquelle konzentrieren muss. Dieses Konzept erfüllen alle Implementierungen und die user experience wurde von vielen Versuchspersonen als positiv beschrieben.

Rekapituliert man die weiteren Studienergebnisse, so wurde gezeigt, dass die Anzahl der zu suchenden Geräusche eine kleinere Rolle auf die benötigte Zeit und Präzision der Lokalisierung eines Geräusches im Raum ausübt, als die Auswahl des zu suchenden Geräusches. Die Auswahl der zu suchenden Geräusche sollte somit in zukünftigen Projekten eine hohe Priorität besitzen.

6.2 Ausblick

Die in dieser Arbeit beschriebene Version der Anwendung ist ein Prototyp zum Testen der Funktionalität der Ortung von Geräuschen mittels Drehen des Smartphones. Jedoch bilden die Erkenntnisse aus diesem Prototyp eine solide Basis, um die Anwendung weiterzuentwickeln und zu erweitern, sodass sie Anwendung schlussendlich in die "Track your Tinnitus" App integrieren werden kann.

Ein wichtiger nächster Schritt zum Verbessern der Anwendung wäre eine erweiterte Gamifikation, um eine Langzeitmotivation und Spielspaß zu gewährleisten. Egal wie viele Vorteile eine mobile Anwendung einem User bietet, sind diese wertlos, falls dieser nicht motiviert ist, die Anwendung zu benutzen. Um die zunehmende Gamifikation zu erzielen stellen sich einige Möglichkeiten dar.

Eine kurzfristige Lösung ist das Hinzufügen weiterer Geräusche mit den passenden Tierbildern, sowie weitere Hintergrundbilder. Diese könnten entweder zufällig oder vom Nutzer selbst ausgewählt, oder mit einer entsprechenden Schnittstelle eventuell vom Anwender selbst integriert werden. Ebenso wären alternative Spielumgebungen eine Möglichkeit die Anwendung zu erweitern. So könnte sich die Suche des Geräusches beispielsweise in einer Stadt abspielen, wobei der Spieler die Richtung erkennen muss, aus welcher ein Krankenwagen anfährt.

Diese Möglichkeiten dienen einer besseren Kurzzeitmotivation. Um eine Langzeitmotivation für die Anwendung zu erschaffen, wäre eine Art Kampagnen-Modus denkbar. Hierbei muss der Benutzer beispielsweise verschiedene Welten, wie Jungle, Savanne, unter Wasser, oder ähnliches bereisen und jeweils passende Tieren oder andere Geräuschquellen fotografieren. Besitzen die Bilder eine gewisse Qualität, so können neue Landschaften freigeschaltet werden.

Ebenso wäre ein Bewertungssystem der Fotos denkbar, beispielsweise mittels Noten, Punkten, oder einer Ingame-Währung. Diese Wertung würde den Benutzer neben der Kampagne auch zu eine Art freies Spiel ermuntern, bei welchem ein Highscore erzielt werden muss. Die Möglichkeiten, Motivation für den Benutzer zu schaffen sind nahezu grenzenlos.

Ein weiterer wichtiger Schritt ist die Verbesserung der Präzision von Ortung und Rotationserkennung. Die in der Android- und iOS-Version verwendete OpenAL API bietet hierfür beispielsweise geeignete Mittel. Durch Feinschliff an den benutzten Methoden für die Wiedergabe der räumlichen Geräusche, sowie durch größere Studien mit ausgeglichenen Gruppen von Tinnitusbetroffenen Personen im Vergleich zu Kontrollgruppen, lässt sich die Ortung der Geräusche sicher deutlich optimieren. Befindet sich die Präzision auf einem angemessenen Niveau, so könnte die Anwendung neben der puren auditorischen Stimulation ebenfalls zur Messung der eigenen Hörleistung genutzt werden. So kann die Abweichung zur Geräuschquelle ähnlich wie in den Logfiles aufgezeichnet werden und dem Anwender anzeigen, wie präzise er Geräusche lokalisieren kann. Wenn man die Idee weiterentwickelt, könnten diese Daten ebenfalls in der "Track

your Tinnitus" App verwendet werden, um eventuelle Schwankungen in einen Bezug zur Tinnitusintensität zu setzen.

Würde man eine Möglichkeit implementieren, mit der der Frequenzbereich des Ohrgeräusches des Benutzers ermittelt werden kann, so könnte die Anwendung sogar zur Behandlung des Tinnitus genutzt werden. Wie bereits bei den Behandlungsmöglichkeiten in Abschnitt 2.1 erwähnt, konnte durch ein sogenanntes "Tailor-Made Notched Music Training", bei dem der Frequenzbereich des Tinnitus aus Musikstücken gefiltert wurde, die wahrgenommene Tinnituslautstärke bei Testpersonen gesenkt werden. Eventuell lässt sich dieser Effekt auch bei den zu ortenden Tönen und den Hintergrundgeräuschen nutzen, sodass die Tinnitusintensität nicht nur kurzfristig durch akute auditorische Stimulation, sondern auch längerfristig durch regelmäßiges "Tailor-Made Notched Music Training" positiv beeinflusst werden kann.

Grundsätzlich lässt sich sagen, dass das Smartphone eine ideale Möglichkeit darstellt, Tinnitusbetroffenen zu helfen. Die meisten Menschen, die ein Smartphone besitzen, tragen dies stets bei sich. Da ein chronischer Tinnitus durchgehend vom Betroffenen wahrgenommen werden kann, muss eine Anwendung, die bei der Bewältigung des Tinnitus helfen möchte, ebenfalls stets erreichbar sein.

Die mobile Anwendung "Track your Tinnitus" bietet einen guten Anfang dabei, dem Betroffenen rund um die Uhr zur Seite zu stehen. Mit zukünftigen Erweiterungen der Anwendung, kann diese Hilfe sogar noch verbessert werden. So könnte dem Tinnituspatienten neben der Erfassung der Tinnitusintensität, auch ein Gehörtraining und kurze spielerische Aufgaben zur Linderung bei akuten Beschwerden zur Verfügung gestellt werden.

Durch Abdecken der weit verbreitetsten Smartphone Betriebssysteme Android und iOS und zukünftig auch Windows Phone und das betriebssystemunabhängige HTML5, wird der erfolgsversprechende Versuch unternommen, der größtmöglichen Anzahl von Menschen, die an chronischem Tinnitus leiden, zu helfen.

Literatur

- [1] BREITHUT, Jörg: *In dieser App kommen Zombies durch den Kopfhörer.* Spiegel Online. <http://www.spiegel.de/netzwelt/games/audio-defence-zombie-arena-fuer-ios-a-1000419.html>.
Version: 2014. – Stand: November 16, 2015
- [2] BUNCHBALL: Gamification 101: An Introduction to the Use of Game Dynamics to Influence Behavior. In: *White paper* (2010)
- [3] GEIGER, Philip ; SCHICKLER, Marc ; PRYSS, Rüdiger ; SCHOBEL, Johannes ; REICHERT, Manfred: Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned. In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*, 2014, 383–394
- [4] HACK, Morgana: *Behandlung des objektiven Tinnitus.* <http://www.tinnitus-mag.de/behandlung-und-therapie-bei-tinnitus/behandlung-des-objektiven-tinnitus/>.
Version: 2012. – Stand: November 16, 2015
- [5] JIN, Craig T. ; CORDEROY, Anna ; CARLILE, Simon ; VAN SCHAİK, Andre: Spectral Cues in Human Sound Localization. In: *Neural Information Processing Systems Conference* (2006)
- [6] KREUZER, Peter M. ; VIELSMEIER, Veronika ; LANGGUTH, Berthold: Chronischer Tinnitus - eine interdisziplinäre Herausforderung. In: *Deutsches Ärzteblatt* (2013), S. 278 – 284
- [7] MARSHALL, Steve: *MP3 Surround.* Sound On Sound. <http://www.soundonsound.com/sos/jan08/articles/mp3surround.htm>.
Version: 2008. – Stand: November 16, 2015
- [8] CREATIVE TECHNOLOGY LIMITED: *OpenAL Programmer's Guide.* https://www.openal.org/documentation/OpenAL_Programmers_Guide.pdf.
Version: 2007. – Stand: November 16, 2015

- [9] DT. GES. F. HALS-NASEN-OHREN-HEILKUNDE, KOPF- UND HALS-CHIRURGIE: *Leitlinie Tinnitus*. <http://www.phoniatrie-paedaudiologie.com/Informationen/HoersturzTinnitus/assets/AWMFonline-Leitlinie%20HNO-Tinnitus.pdf>. Version:2011. – Stand: November 16, 2015
- [10] ULM UNIVERSITY: *Track your Tinnitus - Über uns*. <https://www.trackyourtinnitus.org/de/about>. Version:2013. – Stand: November 16, 2015
- [11] MICROSOFT: *Windows 8 Design and coding guidelines*, 2014
- [12] MICROSOFT: *Applying a 3D Positional Effect to a Sound*. <https://msdn.microsoft.com/en-us/library/bb447687.aspx>. Version:2015. – Stand: November 16, 2015
- [13] MICROSOFT: *AudioListener Class*. <https://msdn.microsoft.com/en-us/library/microsoft.xna.framework.audio.audiolistener.aspx>. Version:2015. – Stand: November 16, 2015
- [14] MICROSOFT: *How to use the combined Motion API for Windows Phone 8*. [https://msdn.microsoft.com/en-us/library/windows/apps/hh202984\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/hh202984(v=vs.105).aspx). Version:2015. – Stand: November 16, 2015
- [15] MICROSOFT: *How to use the Isolated Storage Explorer tool for Windows Phone 8*. [https://msdn.microsoft.com/en-us/library/windows/apps/hh286408\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/hh286408(v=vs.105).aspx). Version:2015. – Stand: November 16, 2015
- [16] MICROSOFT: *Installieren Sie XAP-Dateien von einer SD-Karte auf Ihrem Handy*. <http://www.windowsphone.com/de-de/how-to/wp8/apps/how-do-i-install-apps-from-an-sd-card>. Version:2015. – Stand: November 16, 2015
- [17] MICROSOFT: *Nokia Lumia 730 Dual Sim - Funktionen*. http://www.microsoft.com/de-de/mobile/smartphone-handy/lumia730-dual-sim/technische-daten/#head_the-essentials. Version:2015. – Stand: November 16, 2015
- [18] MICROSOFT: *Nokia Lumia 830*. <http://www.microsoft.com/de-de/mobile/smartphone-handy/lumia830/>. Version:2015. – Stand: November 16, 2015

- [19] MICROSOFT: *Sensors for Windows Phone 8*. [https://msdn.microsoft.com/en-us/library/windows/apps/hh202968\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/hh202968(v=vs.105).aspx).
Version: 2015. – Stand: November 16, 2015
- [20] NASA: *Aircraft Rotations*. <http://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html>. Version: 2015. – Stand: November 16, 2015
- [21] POTISK, Tilen: *Head-Related Transfer Function*. http://mafija.fmf.uni-lj.si/seminar/files/2014_2015/Seminar_Ia_Head-Related_Transfer_Function_Tilen_Potisk.pdf. Version: 2015. – Stand: November 16, 2015
- [22] SCHICKLER, Marc ; PRYSS, Rüdiger ; SCHOBEL, Johannes ; REICHERT Manfred: *An Engine Enabling Location-based Mobile Augmented Reality Applications*. Version: 2015. <http://dbis.eprints.uni-ulm.de/1137/>.
In: *Web Information Systems and Technologies - 10th International Conference, WEBIST 2014, Barcelona, Spain, April 3-5, 2014, Revised Selected Papers*. Springer, 2015 (LNBIP)
- [23] SCHICKLER, Marc ; REICHERT, Manfred ; PRYSS, Rüdiger ; SCHOBEL, Johannes ; SCHLEE, Winfried ; LANGGUTH, Berthold: *Entwicklung mobiler Apps: Konzepte, Anwendungsbausteine und Werkzeuge im Business und E-Health*. Springer-Verlag, 2015
- [24] SCHLEE, Winfried ; SCHECKLMANN, Martin ; LEHNER, Astrid ; KREUZER, Peter M. ; VIELSMEIER, Veronika ; POEPPL, Timm B. ; LANGGUTH, Berthold: *Reduced Variability of Auditory Alpha Activity in Chronic Tinnitus*. In: *Hindawi Publishing Corporation Neural Plasticity 2014 (2014)*
- [25] SCHOBEL, Johannes ; RUF-LEUSCHNER, Martina ; PRYSS, Rüdiger ; REICHERT, Manfred ; SCHICKLER, Marc ; SCHAUER, Maggie ; WEIERSTALL, Roland ; ISELE, Dorothea ; NANDI, Corina ; ELBERT, Thomas: *A generic questionnaire framework supporting psychological studies with smartphone technologies*. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference, 2013*, 69–69
- [26] SCHOBEL, Johannes ; SCHICKLER, Marc ; PRYSS, Rüdiger ; MAIER, Fabian ; REICHERT, Manfred: *Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned*. In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps, 2014*, 371–382

- [27] SCHOBEL, Johannes ; SCHICKLER, Marc ; PRYSS, Rüdiger ; NIENHAUS, Hans ; REICHERT, Manfred: Using Vital Sensors in Mobile Healthcare Business Applications: Challenges, Examples, Lessons Learned. In: *9th Int'l Conference on Web Information Systems and Technologies (WEBIST 2013), Special Session on Business Apps*, 2013, 509–518
- [28] SCHOBEL, Johannes ; SCHICKLER, Marc ; PRYSS, Rüdiger ; REICHERT Manfred: Process-Driven Data Collection with Smart Mobile Devices. Version:2015. <http://dbis.eprints.uni-ulm.de/1136/>. In: *Web Information Systems and Technologies - 10th International Conference, WEBIST 2014, Barcelona, Spain, Revised Selected Papers*. Springer, 2015 (LNBIP)
- [29] SCHWICHTENBERG, Dr. H.: *Erklärung des Begriffs: Isolated Storage*. https://www.it-visions.de/glossar/alle/2619/Isolated_Storage.aspx. Version:2015. – Stand: November 16, 2015
- [30] TINNITRACKS: *Tinnitus-Symptome*. <http://www.tinnitracks.com/de/tinnitus/symptome>. Version:2015. – Stand: November 16, 2015
- [31] TINNITUS-ZENTRUM-SCHWEIZ: *Alles über Tinnitus*. http://www.tinnitus-zentrum.ch/index.php?option=com_content&view=article&id=3&Itemid=9. Version:2015. – Stand: November 16, 2015
- [32] ZHONG, Xiao-li ; XIE, Bo-sun: Head-Related Transfer Functions and Virtual Auditory Display. In: *Soundscape Semiotics - Localization and Categorization* (2014)

A Anhang

A.1 Fragebogen der begleitenden Studie



Universität Ulm
Fakultät für Ingenieurwissenschaften
und Informatik
Institut für Datenbanken und
Informationssysteme

Fragebogen

zur Evaluierung von auditorer Stimulation mithilfe einer Applikation für
die Plattformen Android, iOS, Windows Mobile und mithilfe einer auf
Webtechnologie basierenden Applikation

1. Basisdaten

ID: _____

Alter: _____

Geschlecht: männlich weiblich

2. Haben Sie Tinnitus oder eine andere Erkrankungen bezüglich Ihres Gehörs?

.....
.....
.....

3. Wie viel Erfahrung haben Sie mit mobilen Geräten?

wenig viel
1 2 3 4 5

4. Welches Smartphone verwenden Sie üblicherweise?

.....

5. Wie viel Erfahrung haben Sie mit Videospielen?

wenig					viel
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	2	3	4	5	

Ich wurde vom Versuchsleiter über den Ablauf der Studie aufgeklärt und bin damit einverstanden, dass im Rahmen dieser Studie erhobene personengebundene Daten in anonymisierter Form für wissenschaftliche Zwecke verwendet werden.

Mir ist bekannt, dass ich meine Einwilligung jederzeit ohne Angabe von Gründen und ohne nachteilige Folgen für mich zurückziehen und einer Weiterverarbeitung meiner Daten jederzeit widersprechen und ihre Löschung bzw. Vernichtung verlangen kann.

Datum, Ort, Unterschrift

Name: Martin Weidhaas

Matrikelnummer: 723960

Erklärung

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

Ulm, den

Martin Weidhaas