

Universität Koblenz-Landau Campus Koblenz

FB 4: Institut für Informatik

Betreuer: Prof. Dr. Stefan Müller

Wintersemester 2012/2013

Abgabetermin: 22.03.2013

Bachelor-Thesis



dentAR:

Augmented Reality für den Zahnarzt

Stefan Niederprüm

Computervisualistik

12. Semester

Gartenstraße 8

56070 Koblenz

Matrikelnummer: 207110638

Danksagung

An dieser Stelle möchte ich all denen ganz herzlich danken, die mich beim Entstehen dieser Arbeit begleitet haben:

Allen voran möchte ich mich bei Herrn Prof. Dr. Stefan Müller bedanken, der mich in einem außerordentlichen Maße betreut und unterstützt hat. Vielen Dank!

Dr. med. dent. Daniel Smalla möchte ich danken für viele praktische Tipps, Informationen, Verbesserungsvorschläge, Gebissmodelle und vor allem für seine Freundschaft.

Sinje Diana Meunier möchte ich danken für ihre ununterbrochene Anwesenheit und Fürsorge.

Meinen Freunden möchte ich ganz herzlich für die moralische Unterstützung danken.

Inhaltsverzeichnis

1. Einleitung	S. 3
1.1 Inhalt	S. 3
1.2 Motivation	S. 4
1.3 Methodische Vorgehensweise	S. 4
2. Vorstellung	S.6
2.1 Aktuell	S.6
2.2 Zukunftshardware	S.7
2.3 Systemwahl	S.9
3. Durchführung	S.12
3.1 Vorbereitung	S.12
3.1.1 Marker	S.13
3.1.2 Programmstruktur	S.16
3.1.2.1 Model	S.16
3.1.2.2 View	S.18
3.1.2.3 Controller	S.19
3.1.3 Probleme	S.20
3.2 Programmablauf	S.21
3.2.1 Initiierung der Hauptfunktionen	S.21
3.2.2 Tracker	S.21
3.2.3 Kamera	S.22
3.2.4 Videohintergrund	S.22
3.2.5 Zeichnen: Oberfläche und 3D-Vorbereitung	S.23
3.2.6 Zeichnen: Erkennen der Marker und Anzeigen der 3D-Zähne	S.27
4. Schluss	S.30
4. 1 Probleme und Verbesserungsvorschläge	S.30
4.2 Auswertung	S.33
4.3 Ausblick und Fazit	S.33

Bilderverzeichnis

Quellenverzeichnis

Eigenständigkeitserklärung

1. Einleitung

1.1 Inhalt

Die Überschrift wirbt fast wie ein Slogan für ein Produkt - eine Innovation: die Augmented Reality (AR) für den Zahnarzt. Aber was bedeutet sie überhaupt und welche Realität beschreibt sie? Wie kann der Zahnarzt sie nutzen? Diese AR ist eine Form der Darstellung, bei der die reale Umgebung mit weiteren, virtuellen Daten ergänzt werden kann. Mit Hilfe des Computers werden zusätzliche 2D- oder 3D-Informationen auf einem Bild oder Video angezeigt. Durch diese Ergänzung entsteht eine völlig neue Realitätswahrnehmung, da die reale Welt durch die virtuelle Welt erweitert wird. Diese erweiterte (Englisch: augmented) Realität ist dabei aber nicht mit der virtuellen Realität gleichzusetzen.¹ Die AR ist eine spezielle Form dieser. Sie nutzt deren Eigenschaften und Möglichkeiten und vermischt die Elemente aus ihr, wie beschrieben, mit der Realität. Sie wird daher in den Bereich der vermischten Realität (Mixed Reality) eingeordnet.² Die Augmented Reality besitzt nicht nur aktuell, sondern auch in Zukunft ein großes Potential in verschiedenen Anwendungsbereichen. Diese Technologie kann prinzipiell in allen Bereichen eingesetzt werden. Ein sehr spannendes Teilgebiet mit vielen interessanten Herausforderungen ist die Medizin. Diese Bachelor-Thesis beschäftigt sich mit der Entwicklung eines Programms, welches den Zahnarzt durch die AR bei seiner Behandlung am Patienten unterstützen soll.

Wie könnte man einem Zahnarzt in Zukunft mit technischen Verfahren helfen? Welche nützlichen Informationen könnte ein solches Gerät mit Hilfe der AR anzeigen? Wird die Arbeit des Zahnarztes dadurch erleichtert oder möglicherweise sogar erschwert?

Da die oben genannten Verfahren und die Technik hierfür weder ausgereift noch mir zugänglich sind, wird in dieser Thesis, der Vorgang simuliert und durch die Erstellung einer Applikation, für ein aktuell vorhandenes Apple iOS Gerät beschrieben. Das Gerät

¹ Mehler-Bicher & Reiß & Steiger (2011) S.9 ff.

² Milgram, Takemura, Utsumi, Kishino (1994) S.282 ff.

fingiert eine Brille mit Kamera und Display im Glas, durch die der Zahnarzt erweiterte Informationen angezeigt bekommt. Diese Informationen beziehen sich auf die allgemeinen Patienteninformationen (Name, ID, Zustand der Zähne). Außerdem wird mit Hilfe von Markern für die Augmented Reality ein 3D-Zahngebiss über das vorhandene Patientengebiss gelagert. So können Implantate, Zahnstein, Karies und auch gesunde Zähne visuell hervorgehoben werden.

1.2 Motivation

Bereits während des Studiums hat mich das Themengebiet der AR besonders fasziniert. Die derzeitigen Einsatzmöglichkeiten sind bereits äußerst interessant. Gegeben durch die aktuelle Technologie, aber auch durch die Forschung, beispielsweise im Bereich der Marker (Ziel: markerloses Tracking), ist viel Potential für die Zukunft der AR erkennbar.

Aufgrund meines persönlichen Interesse für Medizin und die medizinischen Visualisierungsmöglichkeiten der Computergrafik sowie der Möglichkeit, in Kooperation mit einem befreundeten Zahnarzt zu arbeiten, wurde schrittweise die Idee für eine Verwendungsmöglichkeit in der Zahnmedizin geboren. Die Relevanz und Anwendungsnähe des Programms für das hypothetische Szenario ist daher gewährleistet.

1.3 Methodische Vorgehensweise

Um eine angemessene theoretische Grundlage zu schaffen, wird zunächst der aktuelle Stand der Technik erläutert, der für dieses Projekt relevant ist. Daraufhin werden mögliche zukünftige Technologien vorgestellt, welche die hypothetische Basis dieser Arbeit darstellen. In dem darauffolgenden Unterkapitel wird die Auswahl der Systeme erläutert, die für dieses Projekt verwendet wurden.

Der Hauptteil beschäftigt sich zunächst mit dem Vorgehen in der Vorbereitungs- und Planungsphase, um daraufhin den Programmablauf der Applikation sukzessiv vorzustellen. Dabei wird auch auf die Probleme eingegangen, die während des Programmierens entstanden sind. In dem reflektierenden Auswertungsteil werden Verbesserungsvorschlägen und Zusatzfunktionen für das geschriebene Programm präsentiert.

2. Vorstellung

Im Folgenden, wird ein Einblick, in die aktuelle Technologie des Smartphones und der Augmented Reality gegeben. Der aktuelle technische Stand soll verdeutlichen, was bereits möglich ist, aber auch abgrenzen, was noch nicht im Bereich des Machbaren liegt (Kapitel 2.1 Aktuell). Dieses Gebiet ist dann Inhalt des darauf folgenden Unterkapitels über die Zukunftshardware (Kapitel 2.2 Zukunftshardware). Um den Rahmen der Arbeit nicht zu sprengen, beschränke ich mich auf folgende Werkzeuge: Brillen mit Display und Kamera, die möglicherweise den Zahnarzt bei seiner Arbeit am Patienten unterstützen können. Da solche Brillen noch nicht für den Endverbraucher marktreif sind, wird in diesem Fall mit einer Applikation auf einem mobilen Gerät (Tablet/ Smartphone) gearbeitet. Daher wird auch die Entwicklung des Smartphones berücksichtigt. Abschließend wird die Auswahl der Systeme für dieses Programm vorgestellt (Kapitel 2.3 Systemauswahl).

2.1 Aktuell

Vor allem in den letzten Jahren und Jahrzehnten hat sich die Technik enorm weiterentwickelt.

Noch vor 20 Jahren war das Handy beispielsweise eine Rarität.³ Doch laut der Marktforschungsfirma Gartner befinden sich diese einfachen Mobiltelefone mittlerweile schon wieder auf dem Rückzug. Der Smartphone-Boom verdrängt die simplen Mobiltelefone Stück für Stück vom Markt. Bereits im Jahre 2012 waren 40 % der verkauften Mobiltelefone Smartphones. Im Jahr zuvor waren es „nur“ 25 %.⁴ Diese Tendenz führt dazu, dass sich die Technologie der Smartphones rasant weiterentwickelt. Die intelligenten Software-Funktionen bieten die Möglichkeit, vielfältige Applikationen zu installieren, die dem Nutzer das tägliche Leben vereinfachen sollen. So wird das Funktionsfeld der mobilen Kleingeräte sukzessiv erweitert. Neue Funktionen werden

³ Steimels (01.06.2012)

⁴ Homepage von Gartner

beispielsweise auch durch Hardwarebauteile, wie Kameras und Sensoren ermöglicht.⁵ Dadurch verfügt das mobile Endgerät über viele Computerfunktionen und kann als kleiner transportabler Computer mit Mobiltelefonfunktion verstanden werden. Von Telefonieren und Internetnutzung, über das Berechnen der 3D-Daten von Spielen und Anwendungen, bis hin zum Fotografieren ist alles möglich.⁶

Genauso wie das Mobiltelefon haben sich auch alle anderen technologischen Bereiche rasant weiterentwickelt. Interessant für dieses Projekt ist vor allem das Segment der Augmented Reality (AR). Die AR ist ein Verfahren, welches das Bild der Wirklichkeit um Zusatzinformationen aus dem Computer ergänzen kann. Beispielsweise im Bereich der Medizin werden Operationsvorbereitung und Planung durch Augmented Reality vereinfacht und optimiert. Mögliche medizinische Anwendungsgebiete werden derzeit an der TU München entwickelt: Ein computergestütztes Visualisierungs- und Navigationssystem mit AR (Narvis) soll das präzise Arbeiten bei minimal-invasiven Eingriffen optimieren. Ziel ist es, dem Chirurg während der Operation die Möglichkeit zu geben, ein 3D-Bild des Inneren des zu behandelnden Körperteils zu sehen, ohne dass dieser auf einen gesonderten Bildschirm schauen muss. Das Bild soll also durch Hinzunahme eines Head Mounted Display (HMD) direkt im Blickfeld des Chirurgen sein. So werden Operationen ermöglicht, die zuvor nicht denkbar gewesen wären. Marktreif wird dieses System aber erst in ein paar Jahren sein.⁷

2.2 Zukunftshardware

Aussagen über die Technologie der Zukunft sind schwer zu treffen. Nachfolgend werden daher nur potentielle Entwicklungen der Smartphones und der Augmented Reality dargestellt.

⁵ Steimels (01.06.2012)

⁶ Rampacher (14.03.2012)

⁷ Homepage der TU München: Research in Medical Augmented Reality - Projekt Narvis

Ein Ende des Smartphone-Booms ist nicht in Sicht.⁸ Die Mutmaßungen über zukünftige Smartphones sprechen von flexiblen Displays⁹ sowie Back-Panel-Displays¹⁰, die beidseitig bedienbar sind. Ich gehe davon aus, dass sich der Trend weg vom Gerät in der Hand entwickeln wird, und hin zu einem Gerät, das mehr Bewegungsfreiheit ermöglicht.



(Bild 2: Google Glass)

Die bereits entwickelten Google Glass stehen kurz vor der Marktreife und sollen noch in diesen Jahr für den persönlichen Gebrauch genutzt werden können.¹¹ Das Google Glass ist ein Brillengestell mit einer kleinen Kamera und einem Display in der oberen äußeren Ecke des Gestells. Es bietet die Funktionen eines Head Up Displays in mobiler Form. So können im Blickfeld des Nutzers, durch das Display, Informationen angezeigt werden. Sprachgesteuerte Befehle ermöglichen es, bei Bedarf, Fotos mit der Kamera zu machen oder andere Aktionen auszuführen, ohne dabei die Hände benutzen zu müssen. Es lässt sich vermuten, dass kommende Projekte diese Technologie verwenden werden, um so mit einer Kombination aus Kameras und Displays den Anwendungsbereich für die Augmented Reality enorm zu erweitern. Dadurch können wiederum viele Möglichkeiten geschaffen werden, um die Lebenswelt des Menschen zu digitalisieren und zu simplifizieren. Die genaue Richtung ist allerdings nicht vorhersehbar: Wird

⁸ Behrens (10.12.12)

⁹ Amour (18.03.13)

¹⁰ Bodonge, (09.03.13)

¹¹ Homepage von Google Glass

tatsächlich eine Brille zur Verwirklichung dieser Technik genutzt? Die Technik schreitet immer weiter fort und visuelle Darstellungen sind bereits auf Kontaktlinsen möglich.¹²

2.3 Systemwahl

Zu Beginn der Erarbeitungsphase wurden die Grundlagen des Augmented Reality ausführlich studiert. Ziel dieser Arbeit ist es, eine Applikation für iOS-Geräte zu entwickeln, die eine Brille mit Kamera und Display imitiert. Deswegen muss ein System gefunden werden, welches dazu in der Lage ist, AR mit Hilfe eines iOS-Gerätes darzustellen.

Auf der Basis dieser Anforderungen kristallisierte sich nach einer längeren Recherche der zur Verfügung stehenden Augmented Reality Kits, das System *Vuforia* von *Qualcomm*¹³, als das adäquate System für dieses Projekt heraus. Dieses System ist kostenlos und kann für nicht kommerzielle Zwecke verwendet werden. Es bietet die benötigten Funktionen von Frame Markern, aber auch von Image Markern. Für ‚*Vuforia von Qualcomm*‘ spricht vor allem die Integration in das gewünschte Betriebssystem iOS. Es bietet zudem eine gute Online-Dokumentation sowie vorgefertigte Testbeispiele, welche die Funktionen der Anwendungsmöglichkeiten präsentieren und erklären. Zum Beispiel wird die Verwendung der Frame Marker bzw. Image Marker durch Videos und Code-Beispiele verdeutlicht. *Qualcomms Vuforia* verwendet zum Rendern OpenGL ES. Nachteilig an *Vuforia* ist, dass die AR-Applikationen zum Testen immer auf dem Endgerät laufen müssen. Eine Simulation in der Entwicklungsumgebung ist nicht möglich. Dies erhöht den Arbeitsaufwand während des Programmierens. Um das 3D-Rendering mit Hilfe einer Engine zu vereinfachen und das Importieren und Verwenden von 3D-Objekten zu ermöglichen, muss eine weitere Schnittstelle integriert werden. *Vuforia* bietet diesbezüglich ein Unity Plug In an. Es ermöglicht eine Integration in Unity 3D. Die Unity 3D Verwendung für iOS-Geräte ist allerdings nur in der Pro Version realisierbar, welche für diese Zwecke zu teuer war.¹⁴

¹² Mehler-Bicher & Reiß & Steiger (2011) S.47 ff.

¹³ Homepage von Qualcomm Vuforia

¹⁴ Homepage von Qualcomm Vuforia

Daher wurde die *NinevehGL 3D Engine*¹⁵ verwendet. Das *NinevehGL* Framework bietet nativ die Möglichkeit zur Realisierung des Augmented Reality Systems für iOS. Dieses Framework ist aktuell in einer Beta-Version verfügbar. Es erweitert die OpenGL ES 2.0 Funktionen und wird in die Apple-Programmier-Umgebung *Xcode*¹⁶ integriert. *NinevehGL* benutzt dabei intern das zur Darstellung von AR gewünschte *Vuforia* von *Qualcomm*. Ein Vorteil von *NinevehGL* dabei ist, die Integrierung der 3D-Rohdaten (Meshes) der Zähne. Diese können direkt aus den 3D-Software-Dateien übernommen werden. Die 3D-Objekt-Dateien (OBJ, Collada) werden importiert und *NinevehGL* passt sie entsprechend an. Das Ändern der 3D-Modelle mit OpenGL wird vom Framework übernommen.¹⁷

Zur Modellierung und Erstellung der einzelnen Zahndaten wurde das Programm ‚Maya‘¹⁸ verwendet. Dieses Programm ist intuitiv bedienbar, kostenlos in der Education-Version und für die Anforderungen dieser Arbeit ausreichend.¹⁹ Als Basis für die 3D-Zahnmodelle wurde das 3D-ZahngGebiss von *VikingDigital* importiert. Es ist ein vollständiges Zahnmodell, welches detailliert gestaltet und zudem erschwinglich ist.²⁰ Der eigentliche Arbeitsaufwand für die Gestaltung und Erstellung der Zähne konnte so minimiert werden.

¹⁵ Homepage von NinevehGL

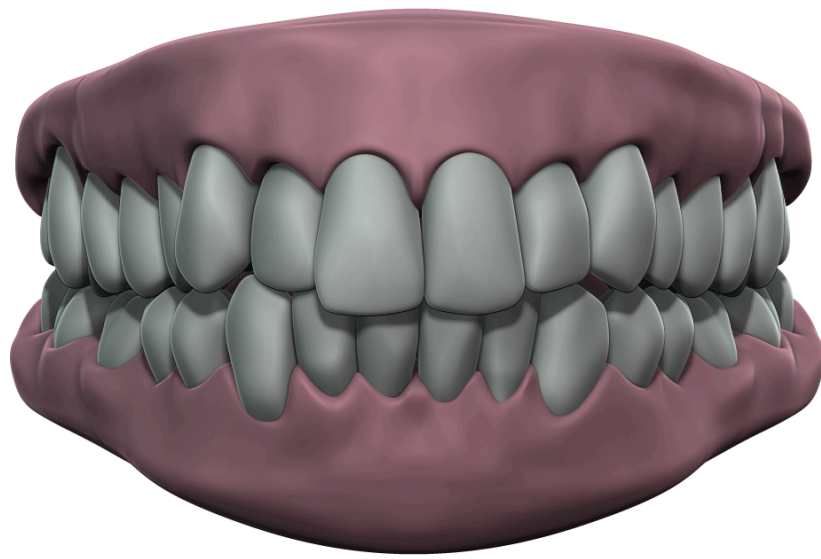
¹⁶ Homepage von Apple Xcode

¹⁷ Homepage von NinevehGL

¹⁸ Homepage von Maya

¹⁹ Homepage von Maya

²⁰ Homepage von Turbosquid: Viking Digital



(Bild 3: 3D-Zahnmodell von VikingDigital)

3. Durchführung

Das folgende Kapitel beschäftigt sich mit dem wichtigsten Teil der Bachelor-Thesis: Die Beschreibung der praktischen Erarbeitung des Programms. Dabei werden zunächst die Tätigkeiten während der Vorbereitungsphase und Planung dargestellt. Für die Programmstruktur wird das MVC Pattern (Model-View-Controller Pattern) als Basis benutzt, wodurch das Programm dreiteilig strukturiert wird. In dem sogenannten Model-teil werden die Daten und Objekte definiert. Im View-Teil wird die Benutzeroberfläche mit den Ansichten und Bedienungselementen entwickelt. Der Controller-Teil beschäftigt sich mit der Funktionsfähigkeit der Benutzeroberfläche und dem Verbinden der Daten aus dem Modell mit den dazugehörigen Ansichten. Daraufhin wird der Programmablauf der Applikation beschrieben. Die in der Struktur genannten Elemente werden dabei chronologisch initiiert.²¹

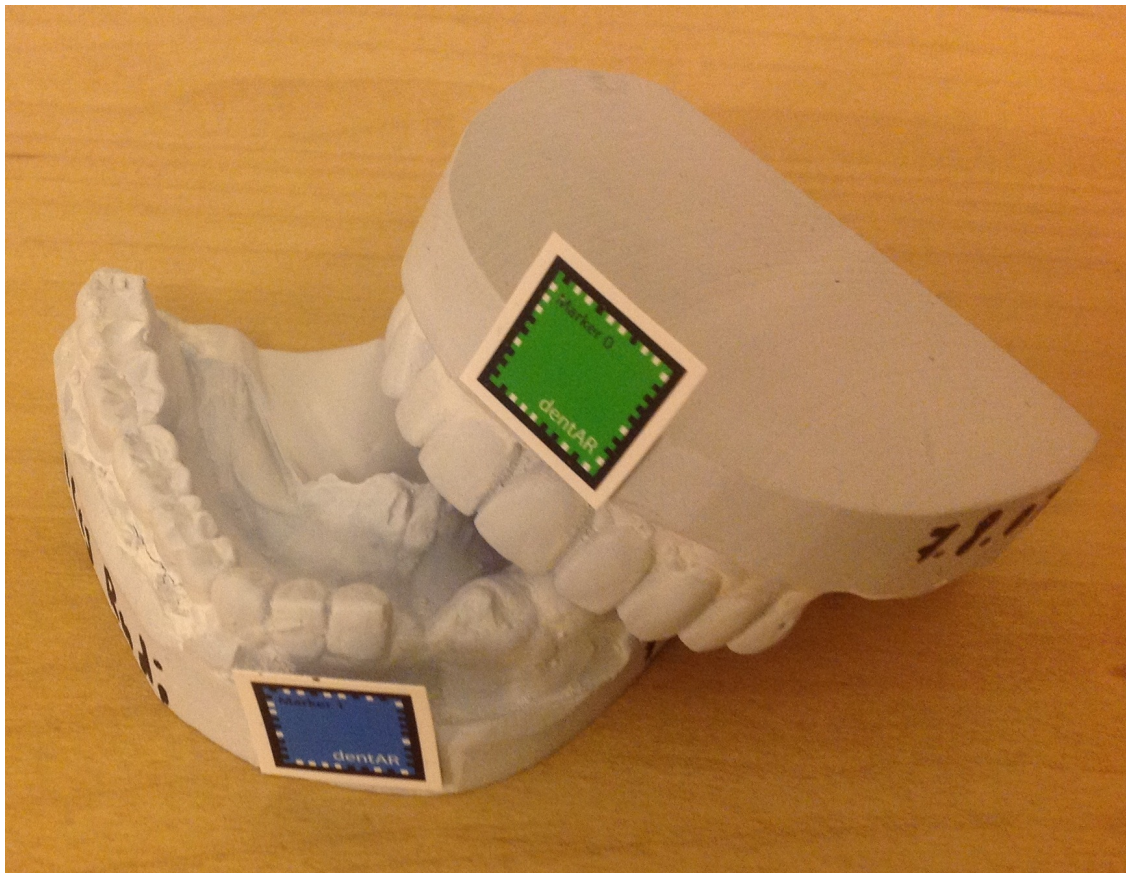
3.1 Vorbereitung

Wie bereits in dem Kapitel 2.3 ‚Systemwahl‘ erwähnt wurde, läuft der Programm-Code nur auf einem iOS-Gerät und nicht auf dem PC selbst. Das iOS-Gerät muss über einen Apple Developer Account registriert werden, um das Testen der Applikation zu ermöglichen. Die benötigten Frameworks können nun installiert und deren Funktion getestet werden. Dazu wurde von mir ein eigenes Beispielprogramm geschrieben, das die Marker aus dem vorhandenen Testbeispiel erkennt und ein komplettes 3D-Gebiss über dem Marker anzeigt. Die Herausforderung besteht nun darin, ein eigenes vollständiges Testszenario zu entwerfen, um die grundlegende Funktion zu ermöglichen: Ein Gebiss, mittels eigener vordefinierter Marker darzustellen. Wie kann nun ein Augmented Reality System erstellt werden, mit dem visuelle Zähne auf vorhandene Plastikzähne bzw. reale Zähne, überlagert werden können? Welche Daten, Objekte und Methoden sind dafür von Bedeutung und müssen daher eingerichtet werden?

²¹ Gamma & Helm & Johnson & Vlissides (1994) S. 4 ff.

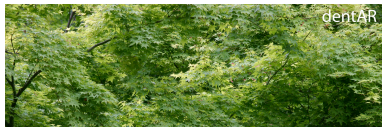
3.1.1 Marker

In dem Zielszenario (Praxis) schaut der Zahnarzt dem Patienten in den Mundraum. Da der Mundraum sehr klein ist, dürfen die Marker in Relation dazu nicht zu groß sein. Damit der Arzt beim Arbeiten nicht durch die Marker gestört wird und die Marker nicht durch den Zahnarzt verdeckt werden, sollten sie außerhalb des Mundraums platziert werden. In einem nächsten Schritt wurde die Entscheidung über Art und Befestigung der Marker getroffen. Für 3D und Augmented Reality ist die Position der Marker wichtig, da die Daten in Relation zu den Markern angezeigt werden sollen. Auch spielt die stabile Position der Marker eine große Rolle, um eine stetige und verlässliche Erkennung dieser zu gewährleisten. Andernfalls müsste bei jeder Verschiebung der Marker eine erneute Skalierung und Einrichtung erfolgen. Da die Handlung nur prototypisiert wird und folglich Plastikzähne verwendet werden, können die Marker dort fix befestigt werden.



(Bild 4: Gipszahngebiss mit Markern)

Dies erleichterte die Arbeit an dem Projekt. Um eine realitätsnahe Verwirklichung zu ermöglichen, fiel die Wahl auf kleine rechteckige Image Marker, die im Gesichtsfeld nahe des Mundes positioniert werden konnten.



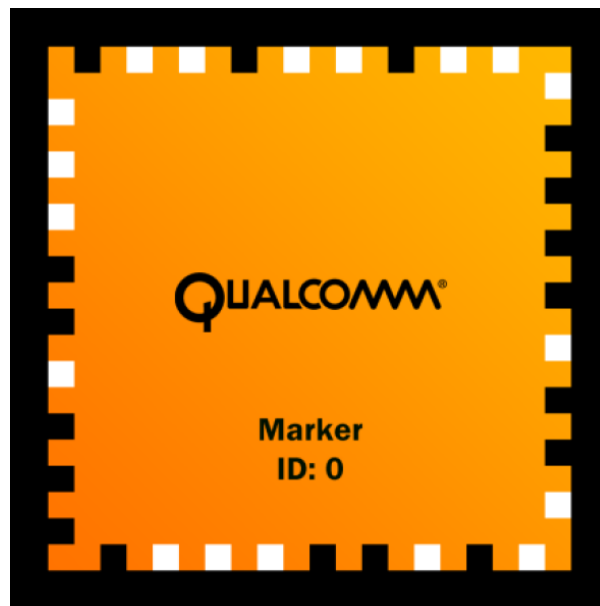
(Bild 5: Selbst erstellte Test Image Marker)

Die Image Marker werden dabei im Online Tool von *Vuforia* erstellt. Man kann hier Bilder hochladen und die Breite angeben. Die Bilder sollten möglichst kontrastreich und ungleichmäßig sein, sodass markante Stellen vorhanden sind. Die auffälligen und charakteristischen Stellen im Bild werden in einer Datei gespeichert. Das Online Tool von *Vuforia* bewertet die Eignung der Marker und gewährleistet somit deren Funktion. Durch die große Zahl an Auffälligkeiten im Bild und deren verteilten Positionen zueinander, ist das Tracking auch durch eine teilweise Verdeckung der Marker möglich. Aufgrund der Tatsache, dass die verwendeten Image Marker bei schlechten Lichtverhältnissen und kleiner Ausführung jedoch nur suboptimale Ergebnisse erzielten, wurden in der späteren Erarbeitungsphase doch Frame Marker verwendet. Hinzu kommt, dass die Vorteile der Imagemarker, wie die Überdeckungsmöglichkeit durch ihre Größe kaum verwendbar sind. Frame Marker sind meist quadratische Marker die einen schwarzen Rand besitzen, an dessen Innenseite eine digitale ID codiert ist. Dieser Rand wird zur Identifikation des Markers genutzt. Der Innenteil ist frei wählbar und dient zur optischen Unterscheidung der Marker durch den Benutzer. Bei Verwendung der Marker ist darauf zu achten, dass es einen kontrastreichen Rand um den eigentlichen Rand gibt, damit dieser als solcher erkannt werden kann. *Vuforia* kann 512 verschiedene Frame Marker unterscheiden.²² ²³ Eigens ausgeführte Tests zeigten, dass diese Frame

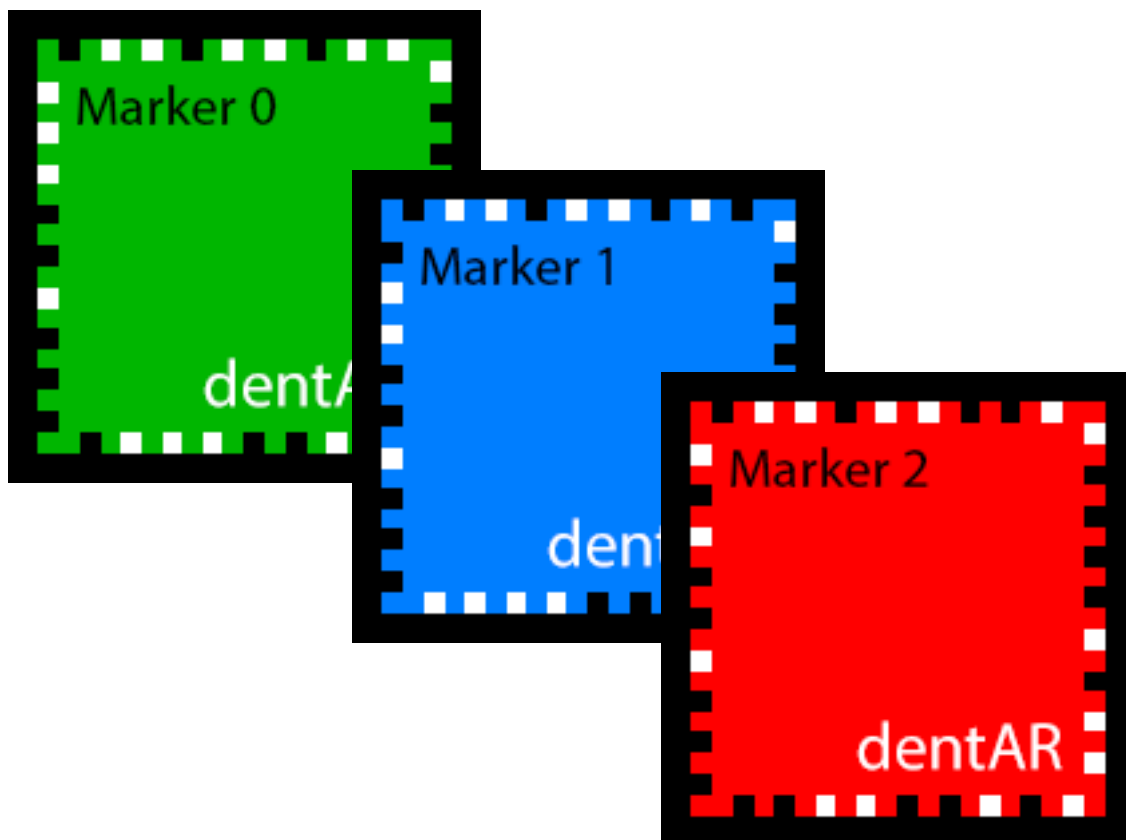
²² Mehler-Bicher & Reiß & Steiger (2011) S.34

²³ Homepage von Vuforia

Marker auch in einer kleinen Ausgabe (15 mm x 15 mm) ordentliche Ergebnisse liefern und somit für diese Zwecke genutzt werden können.



(Bild 6: Original-Vuforia Frame Marker)



(Bild 7: Selbst erstellte Frame Marker)

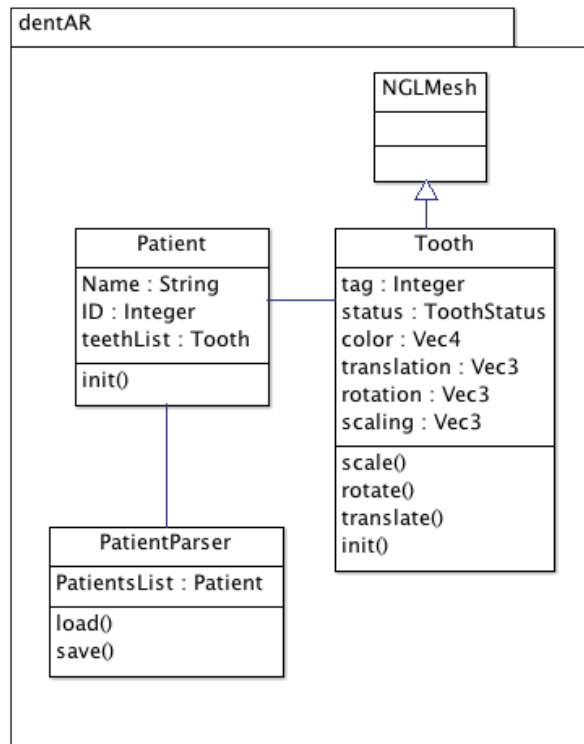
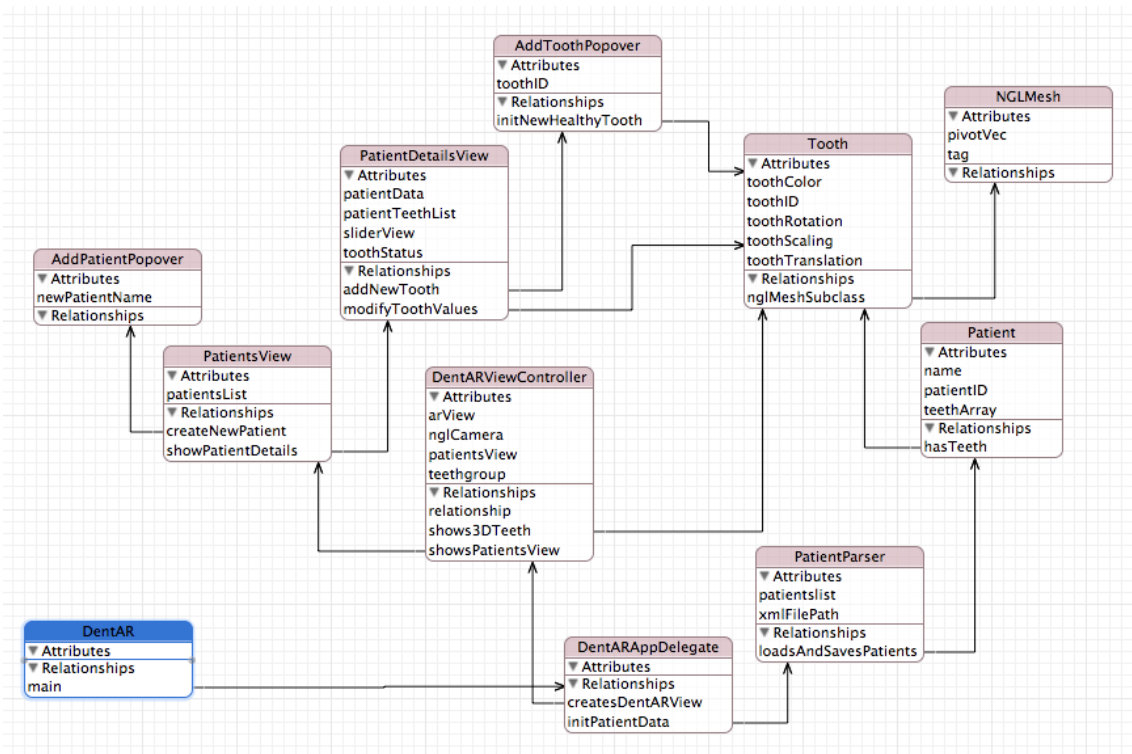
3.1.2 Programmstruktur

Zur Verwirklichung der Applikation verwendet man als Struktur das bereits erläuterte MVC-Konzept. Folgend wird die Implikation des Programms in dieses Modell dargestellt.

3.1.2.1 Model

Es werden Instanzen für die Model-Daten benötigt, die lokal geladen und gespeichert werden können. Zum Verwirklichen der Objekte und Methoden werden nun die zahnarztbezogenen Daten implementiert. Eine XML-Datei in der Applikation simuliert die vorhandene Datenbank des Zahnarztes. Neben Patienten mit ihren spezifischen Identifikationswerten (Name und ID) müssen auch deren Zähne als Objekte vorhanden sein. Da nicht jeder Zahn bei jedem Patienten gleich ist, sind wiederum eigene Werte zur genauen Positionierung, Skalierung und Orientierung sowie zum Status notwendig. Anhand des Status wird dem Arzt farblich signalisiert, wie die Beschaffenheit der Zähne ist. Dabei wird zwischen gesund (weiß), kariesbefallen (blau), zahnsteinbefallen (gelb), karies- und zahnsteinbefallen (grün) und Implantat (grau) unterschieden. Der jeweils ausgewählte Zahn wird durch eine violette Farbgebung hervorgehoben. Mittels eines Parsers²⁴ werden die Informationen aus der XML-Datei geladen und nach möglichen Veränderungen in dieser wieder gespeichert. Die Zahnobjekte sind dabei von 3D-Mesh-Instanzen des NinevehGL Framework abgeleitet. Von dem bereits genannten 3D-Gebiss wurden einzelne 3D-Dateien erstellt, die nur den jeweiligen Zahn repräsentieren. Diese 3D-Dateien werden den passenden Zahnobjekten während der Initiierung zugeordnet. Bild 8 zeigt die Model-Objekte und die ungefähre Struktur der Applikation. Die Beziehungen der einzelnen Elemente werden skizzenhaft dargestellt. Im Laufe des Entwicklungsprozesses wurden in der anfänglichen Skizze einige Funktionen entfernt und andere hinzugefügt.

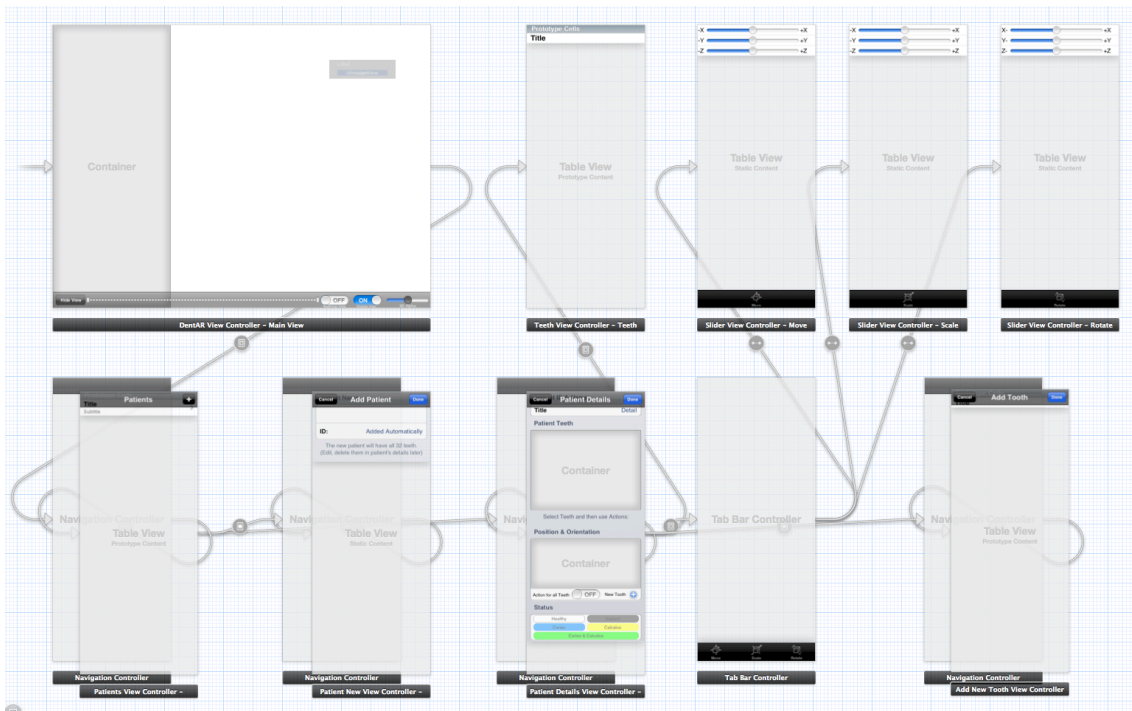
²⁴ Komponente zum Aufteilen der XML Datei und Zuordnen von dessen Elemente zu den Variablen



(Bild 8: Programmstruktur & Datenmodellierung)

3.1.2.2 View

Der Aufbau der Benutzeroberfläche ist so gestaltet, dass der Zahnarzt den Patienten aus der Datenbank auswählt und seine Behandlung beginnt. Zusätzlich werden ihm durch die Applikation die patientenbezogenen Informationen angezeigt. Außerdem werden im Mundraum des Patienten die virtuellen 3D-Objekte eingeblendet. Die dazu benötigte Benutzeroberfläche mit den einzelnen Ansichten (Views) wurden in Xcode im Storyboard modelliert. In der Planungsphase wurde deutlich, dass die Applikation auf einem kleinen Bildschirm zu Problemen führen kann. Die vorgegebenen standardisierten Objekte auf der Benutzeroberflächen für die Interaktionsmöglichkeiten in iOS - wie Buttons, Listen und Slider - hätten zuviel Platz auf dem Display eingenommen. Auch ist zu beachten, dass das Blickfeld des Zahnarztes schon durch das kleine Display eingeschränkt wird. Ein ständiger Wechsel zwischen den Einstellungen, dem Editierungsmodus und der Live-Ansicht (der AR View) war nicht gewünscht und mögliche Überblendungen der Augmented Reality Ansicht mit Popup-Abfragen zum Speichern/Ändern oder den Elementen der Oberfläche wäre nicht praktikabel. Das bedeutet nicht, dass die Funktionen und Möglichkeiten auf dem Smartphone (oder einem anderen Gerät mit kleinem Display) nicht realisierbar sind. Dennoch ist die Darstellung auf dem iPad (Tablet) größer und daher für dieses Programm besser geeignet. Für die genannten Objekte aus dem Datenmodell müssen Ansichten zur Anzeige als Listen erstellt werden. Das ist notwendig, um eine realitätsähnliche Umgebung zu gestalten, wie sie in einer Art Karteikartenverwaltung vorhanden ist. Außerdem werden dem Zahnarzt Statusfelder, Aktion-Buttons, Editierungsmöglichkeiten und Informationsfelder angezeigt. Alles sollte dabei neben bzw. über der AR View angezeigt werden, ohne die Sicht darauf zu klein werden zu lassen, sodass die eigentliche Hauptaufgabe - das Implementieren der Augmented Reality Funktion auf das Display bestehen bleibt.



(Bild 9: Storyboard)

Dieses Bild zeigt das Storyboard mit allen einzelnen Views der Applikation. Es ist durch Skizzen über Einsatzmöglichkeiten und mögliche Anzeigen der Patienteninformation entstanden. Dabei sollten, wie erwähnt die 3D-Zähne nicht verdeckt werden und das Modellieren bzw. Interagieren der vorhandenen Daten leicht möglich sein.

3.1.2.3 Controller

Um die Daten in der View zuzuordnen, werden Controller für diese erstellt. Für jede View wird dabei ein eigener View-Controller benötigt, welcher die Inhalte beim Starten der Ansicht passend lädt. Dadurch werden weitere View-Controller erstellt und als Variable verwaltet. Deren Inhalte können nun korrekt an die Ansicht weitergegeben werden. Um die Daten zwischen den Views zuzuordnen und weiterleiten zu können, besitzt jeder Controller einen ‚Beauftragten‘, der Eigene und fremde Daten delegiert und die Kommunikation mit anderen Views und deren Controller ermöglicht. Somit erkennt die Patienten-Detail-Ansicht, welcher Patient ausgewählt wurde oder welcher Zahn in der Liste jetzt der Ausgewählte ist. Aber auch der Weg von der Zahnliste zur 3D-Darstellung in der AR View und anders herum muss funktionieren. Die korrekten Werte

müssen angezeigt und aktualisiert werden. Wurde z.B. ein Zahn (also das 3D Mesh) direkt über die AR View mit einer Touchgeste selektiert, dann muss auch die Zahnliste des aktiven Patienten und dessen aktiver Zahn erneuert werden. Grundlegend muss hier immer auf die korrekte Initiierung der Objekte und Variablen geachtet werden, sonst wird schnell ein Array mit 0-Objekten verwendet oder es entsteht ein Speicherzugriffsfehler. Auf die grundlegenden und einzelnen Funktionen der Controller wird im Abschnitt Programmablauf genauer eingegangen.

3.1.3 Probleme

Während der Implementierungsphase entstanden unter anderem folgende Probleme:

Zunächst wurden die Zahn-Meshes nicht richtig importiert. Später wurden die Alpha-Werte der RGBA-Farbwerte, durch Fehler beim XML-Import auf „0“ gesetzt und vollkommen transparent dargestellt dargestellt. Dies hatte zur Folge, dass sie im Hintergrund, aber auf dem Bildschirm nicht angezeigt werden konnten. Als problematisch erwies sich die Ansicht für den Video-Inhalt, die den ganzen Bildschirm verdeckte und die weiteren Bedienungselemente nun nicht mehr im Sichtfeld waren. Bei dem Versuch, alle Ansichten korrekt darzustellen, brach z.B. die Initiierung der von *Vuforia* benötigten Daten für die AR ab, da nicht die korrekte Ansicht für den Video-Inhalt und zum Rendern der Daten zugeordnet werden konnte. Letztendlich ließen sich die Programmierschwierigkeiten durch intensive Fehlersuche beheben. Häufig ergaben sich aus vermeintlich kleinen Problemen schwerwiegende Folgefehler. Ein wichtiger Punkt war das richtige Initiieren der Ansichten für das Benutzerinterface und deren Zuordnung, mit Beachtung der Instanzmethoden von Oberklassen. Auch die adäquate Zuweisung der Ansichten durch den Benutzeroberflächen-Gestalter von *Xcode* für das Storyboard ist erforderlich. Durch das Erstellen aller Views und passendes verlinken dieser im Storyboard, werden im Programm keine expliziten Aufrufe und Erstellungen von *Subviews* benötigt. Das Anwendungsfenster und alle Unterfenster werden dabei automatisch erzeugt.

3.2 Programmablauf

Dieses Kapitel beschreibt die Implementierungsschritte anhand der Objekt- und Methodenaufrufe in der Applikation. Die in dem Kapitel 3.1.2 ‚Programmstruktur‘ genannten Bausteine werden möglichst in der Reihenfolge des tatsächlichen Aufrufens in der Applikation erläutert.

3.2.1 Initiieren der Grundfunktionen

Das Starten der Applikation aktiviert die Main Methode. Diese befiehlt dem Hauptbeauftragten (dentARAppDelegate) der Applikation, die Hauptansicht mit ihrem Haupt-Controller dentARViewController als Root-View zu deklarieren. Außerdem lädt er beim Initiieren über den Parser die Patientendaten aus der XML-Datei. Diese werden an den Controller und dessen Liste der verfügbaren Patienten weiter gegeben. Der HauptViewController startet nun seine entsprechende Ansicht. Hier werden einerseits die Elemente aus dem Storyboard mit der Position und Anordnung geladen und andererseits die weiteren Applikationsgrundlagen für *Vuforia* und *NinevehGL* initiiert. *Vuforia* benötigt hierfür zum ordnungsgemäßen Ablauf einige Funktionen, die das Programm beim Initiieren nach und nach aufruft. Zunächst definiert man, wie groß der Bereich ist, der für die AR genutzt werden soll. Die durch den ViewController geladenen Views müssen richtig zugewiesen sein und eventuell vorhandene Instanzmethoden der Oberklasse geladen werden. Wie bereits in Kapitel ‚3.1.2.4 Probleme‘ beschrieben, kann sonst die Initialisierung der AR mit einer Fehlermeldung abbrechen.

3.2.2 Tracker

Zur Darstellung der erweiterten Informationen müssen die Marker beschrieben werden die später im Bild gesucht werden sollen. Dafür werden für die Frame Marker die sogenannten ‚Marker Tracker‘ eingerichtet. Die Tracker für den jeweiligen Marker werden dabei aus den *Vuforia* internen Frame Marker IDs und einem selbst definierten Vector,

der die Größe der Marker angibt, erzeugt. Wie schon in der Vorbereitung beschrieben, benutzt die Applikation Frame Marker. Die notwendigen Daten, für die am Anfang verwendeten und getesteten Image Marker, werden trotzdem geladen. Über deren Konfigurationsdatei wird die Markergröße und der Markername beschrieben, sowie der speziell für die Images charakteristische Datensatz verlinkt. Durch explizites Setzen des Markertyps sind die Image Marker aber im weiteren Verlauf ohne Funktion.

3.2.3 Kamera

Der Tracker wird je nach Markertyp gestartet und das Objekt wird für die *Vuforia*-Kamera initialisiert. Das Gerät liefert dazu die benötigten Kalibrierungsdaten und erstellt die Kamera-Instanz. Mit den Daten der Kamera wird die Projektions-Matrix beschrieben, welche dessen Kalibrierung nutzt. Zusätzlich wird die Projektionsmatrix durch Setzen der Near- und Far-Plane- Werte definiert.²⁵ Die Kamera-Funktion und deren Starten und Stoppen sind Teil der *Vuforia*-Initialisierung. *Vuforia* ermöglicht es aber eine individuelle Kamera, und somit eine eigene OpenGL-Projektions-Matrix zu verwenden. Durch die Integration von *NinevehGL* und dessen Renderfunktionen wird, nachdem die View des HauptViewController geladen wurde, eine *NinevehGL*-Kamera (NGLCamera) Instanz erzeugt. Die Kamera sitzt dabei im Punkt (0,0,1) und schaut entlang der negativen Z-Achse. Die Größe des verwendeten Bildschirmbereichs bestimmt die View, die im Storyboard dafür festgelegt wird.²⁶

3.2.4 Videohintergrund

Wie im Zukunftsszenario erwähnt, soll der Bildschirm das Sichtfeld des Arztes sein und durch ein Zusatzdisplay an der Brille sollen nur die erweiterten Informationen angezeigt und aufs reale Bild (auch Sicht des Arztes) überlagert werden. Für die Videoüberlagerung im Hintergrund wird die Größe und Position ebenfalls durch die *NinevehGL*-View begrenzt. Durch eine leichte Transparenz der übrigen

²⁵ Homepage von Vuforia Developer

²⁶ Homepage von NinevehGL Reference

Bedienungselemente und Anzeigen füllt das Video den Bildschirm komplett aus und ermöglicht eine überlagerte Darstellung.

3.2.5 Zeichnen: Oberfläche und 3D-Vorbereitung

Im Storyboard wurde jeweils der passende View-Controller für die einzelnen Elemente angegeben. Die dazugehörigen Instanzen werden immer erst beim Aufruf erstellt. Das Starten des Haupt-View-Controllers führt, nachdem dessen Ansicht geladen wurde, automatisch die Erstellung der benötigten weiteren Ansichten durch. Zur Darstellung der erstellten Patientenliste wird der Patienten-View-Controller deklariert und als Tabelle angezeigt. Pro verfügbaren Patienten wird eine Zeile erzeugt. Jede Zeile repräsentiert die aus dem Datenmodell bekannten Patientenobjekte mit allen Variablen. Nach der Auswahl eines Patienten, welche durch einen Touch auf die Zeile geschieht, werden die Patientendetails angezeigt.



(Bild 10: Benutzeroberfläche der Patientenliste)

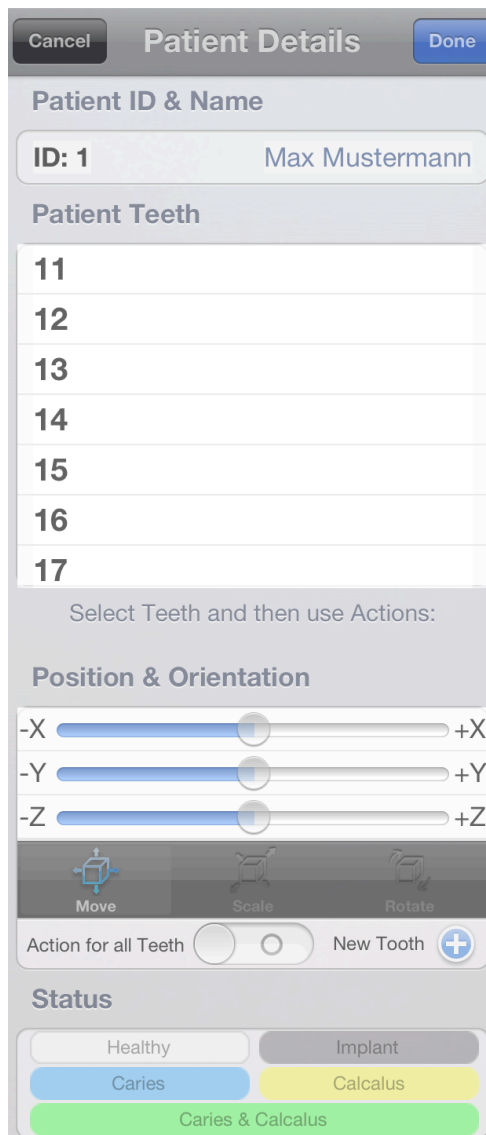
Währenddessen wird in dem Haupt-View-Controller der ausgewählte Patient als aktiv markiert und dessen virtuelles Zahnmodell initiiert. Dieses wird einer 3D-Gruppe hinzugefügt. Die Gruppe dient als Hilfsmittel. Durch sie hat man die Möglichkeit alle Zähne des Patienten auf einmal auszuwählen und zu ändern. Die Zähne besitzen die von ihrer zugehörigen 3D-Datei importierten Eigenschaften bezüglich der Polygonanzahl und -position zum Ursprung, sowie die Standardgröße. Die Zähne werden nun für die Anwendung passend im Raum positioniert. Die Pivot²⁷-Werte, die in Maya gesetzt wurden, werden aktuell nicht übernommen. Mit Hilfe des Pivot-Vectors kann schon im 3D-Programm der Fixpunkt für die Rotation gesetzt werden. Somit müssen die 3D-Objekte im Programm vor deren Modifikation, nicht wie aus OpenGL bekannt, in den Ursprung und danach wieder zurück an ihren Herkunftsort verschoben werden. Der Pivotpunkt wird beim Erstellen der Zahn-Objekte jedoch auf den Ursprung gesetzt. Die Betaversion von Nineveh verfälscht vermutlich die Pivot-Werte. Damit sich die Zähne auch einzeln ausgewählt beim Rotieren korrekt verhalten, müssen diese Werte manuell angepasst werden. Über eine Liste werden jedem Zahn die entsprechenden Werte aus dem 3D-Programm zugewiesen. Dadurch verändert sich jedoch auch die Position der Zähne und sie werden auf den Ursprung gesetzt. Damit sie aber in der Grundposition des Gebisses an ihrer richtigen Stelle liegen, werden die Zähne der Gruppe wieder um den Pivot-Vector verschoben. Dadurch kann der Zahn einzeln korrekt um den eigenen Mittelpunkt rotieren. Die 3D-Gruppe kann im Gesamten weiterhin angemessen skaliert und rotiert werden. Die Gruppe wird dann dem *Nineveh*-Kamera-Objekt hinzugefügt.

Zur späteren Skalierung eines einzelnen Zahns müssen ebenfalls weitere Einstellungen vorgenommen werden. Unabhängig davon an welcher Position sich das 3D-Zahnobjekt befindet, wird intern die Position verwendet, die die 3D-Datei vorgibt. Leider stimmen Pivot-Punkt und Fixpunkt zur Skalierung nicht überein. Dies kann weder durch das Verschieben in den Ursprung noch durch das Setzen des Pivot-Vectors auf 0 gelöst werden. Zur Rotation wurde der Vector für jedes Zahn-Objekt noch einzeln manuell gesetzt und konnte diese so beeinflussen. Die *NinevehGL*-Skalierung verwendet mutmaßlich aber weiterhin den Ursprung der Weltkoordinaten aus dem 3D-Programm.

²⁷ Erklärung Pivot: Dreh- und Mittelpunkt

Exakte Exportierungs-Optionen aus dem 3D-Programm, welche diese Fehler verhindern, wurde nicht gefunden. Damit die Skalierung dennoch korrekt ausgeführt werden kann und nicht durch den Abstand zum „Ursprung“ verzerrt wird, verwendet man auch hier den manuell gesetzten Pivot-Vector. Dieser stellt im Standardgebiss die Position im Gebiss zum Ursprung dar. In der Applikation kann der Zahn für jede Achse im Intervall [0,2] skaliert werden. Der entsprechende Faktor muss dann auch rückwirkend auf den Pivot-Vector angewandt werden. Die Werte der drei Achsen können je nach Faktor doppelt so groß wie die vorherigen Werte oder gleich null sein. Um der Verzerrung entgegenzuwirken, muss eine Verschiebung der Positions-Werte durchgeführt werden. Bei Verkleinerung des Objekts müssen diese erhöht und bei Vergrößerung verkleinert werden. Daher wird während der Skalierung die Position der Zähne durch Modifizieren mit Hilfe der Positions- und Pivot-Werte bestimmt. Diese Werte negieren dabei, den durch die Skalierung entstandenen Unterschied zum Original-Wert der jeweiligen Achse. Die neue Position wird letztendlich durch 3 Werte bestimmt: den alten Wert vor der Skalierung, dem original Pivot-Wert und dem skalierten Pivot-Wert. Da der Pivot-Vector den ursprünglichen Ort im Gebiss angibt und dieser Ort zur Skalierung verwendet wird, muss dieser skalierte Wert vom veränderten Wert der Position abgezogen werden (also: Original + Pivot - Pivot * Skalierung). Dadurch wird der Veränderung der Position, durch Negierung der Skalierung für die Achsen, entgegengewirkt. Der Zahn bleibt nun bei veränderter Größe an der selben Stelle.

Zur Anzeige der Patientendetails wird der Detail-View-Controller erzeugt. Dieser bietet die Ansicht zur Anzeige der Patientendaten, die benötigte Zahn-Liste (der Zähne des Patienten) und deren Eigenschaften werden dargestellt. Die Bedienelemente (Slider und Buttons) zum Manipulieren für Position und Größe der Objekte werden ebenfalls angezeigt. Die dort veränderten Werte werden folglich in den Variablen zwischengespeichert. Erst wenn man in den Details auf „Done“ (Speichern) drückt, werden sie auch in der XML-Datei (Datenbank) gespeichert, sodass sie beim nächsten Aufruf in der gewünschten Form erscheinen.

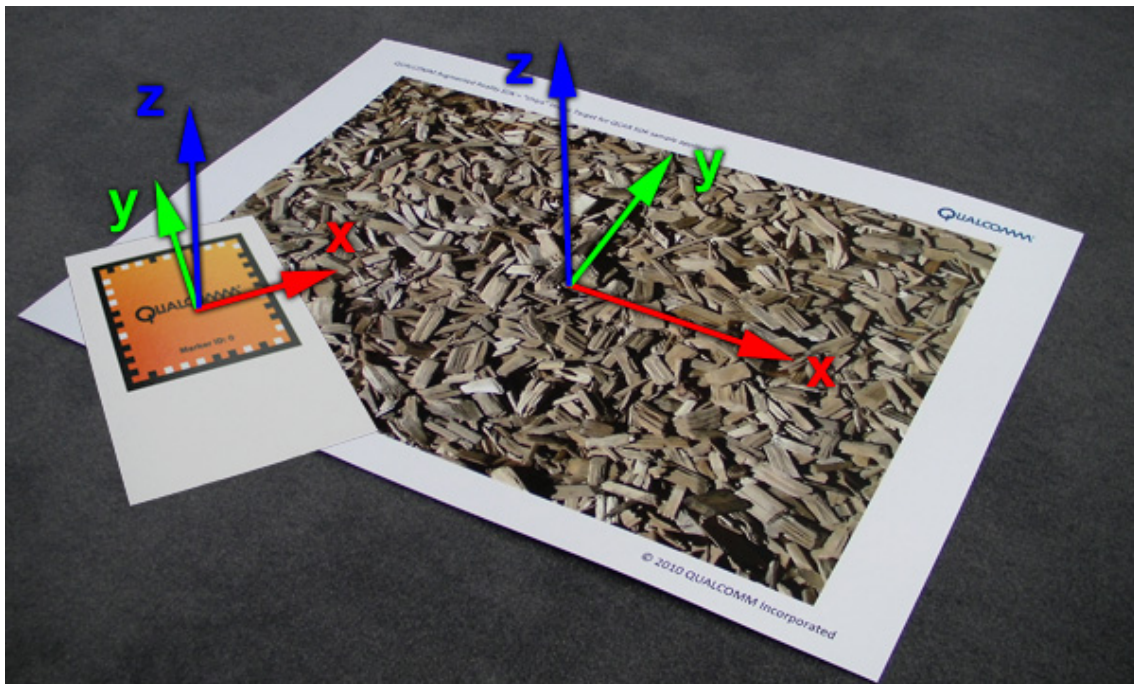


(Bild 11: Benutzeroberfläche der Patientendetails)

Die *Nineveh*-Zeichenmethode für die AR-View im Haupt-View-Controller wird, während die Applikation läuft, ständig aufgerufen, und sucht kontinuierlich nach vorhandenen Markern im (Video-)Bild. Die zuvor gesetzte Konfiguration der Maße und Kalibrierung des Videohintergrunds wird nun auf dem Bildschirm angezeigt. Die 3D-Modelle des aktiven Patienten werden jedoch ausgeblendet, da diese nur angezeigt werden sollen, wenn ein Marker im Bild entdeckt wird.

3.2.6 Zeichen: Erkennen der Marker und Anzeigen der 3D-Zähne

Sobald ein Marker von der Kamera erkannt wird, wird dessen eigene Position und Orientierung (Pose) ermittelt und diese als Matrix an das Programm übergeben. Die Pose der Marker ist eine Matrix, die die Lage zur Kamera angibt. Sie wird bei *Vuforia* inter als 3x4 Matrix beschrieben.²⁸

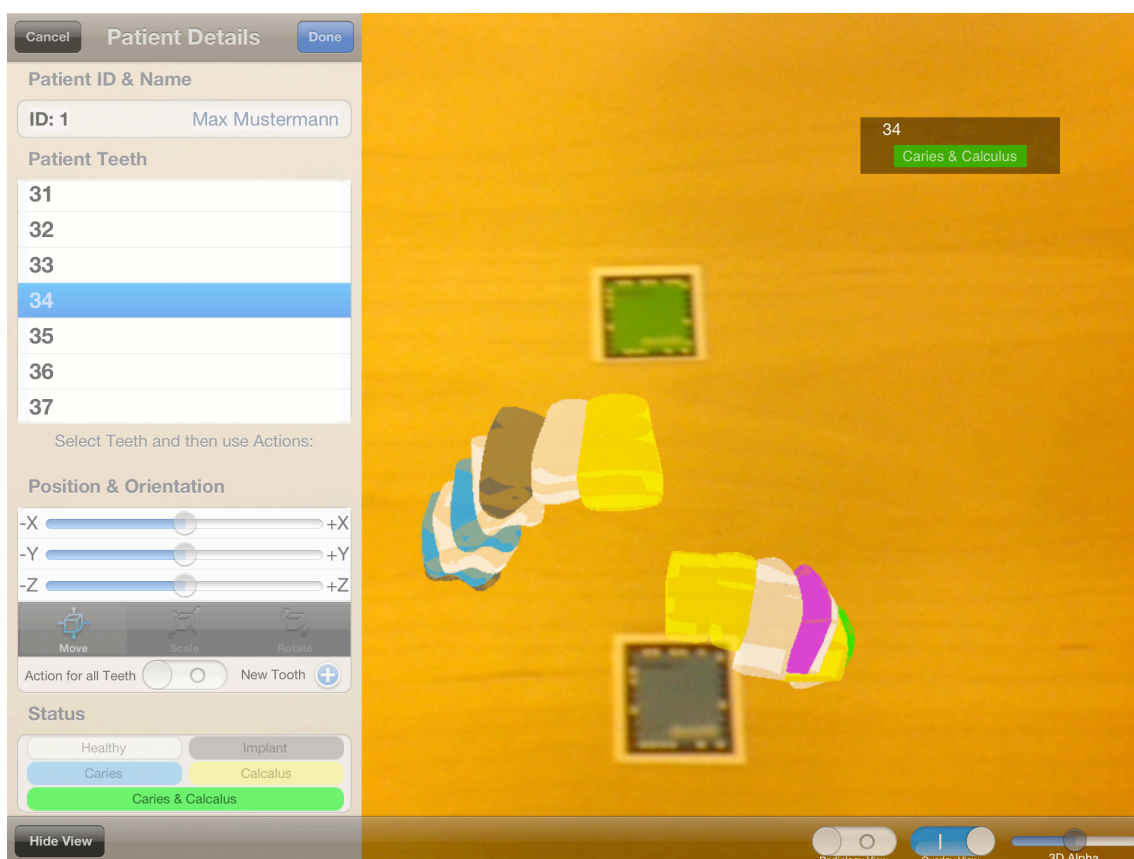


(Bild12: Koordinatensystem der Pose-Matrix)

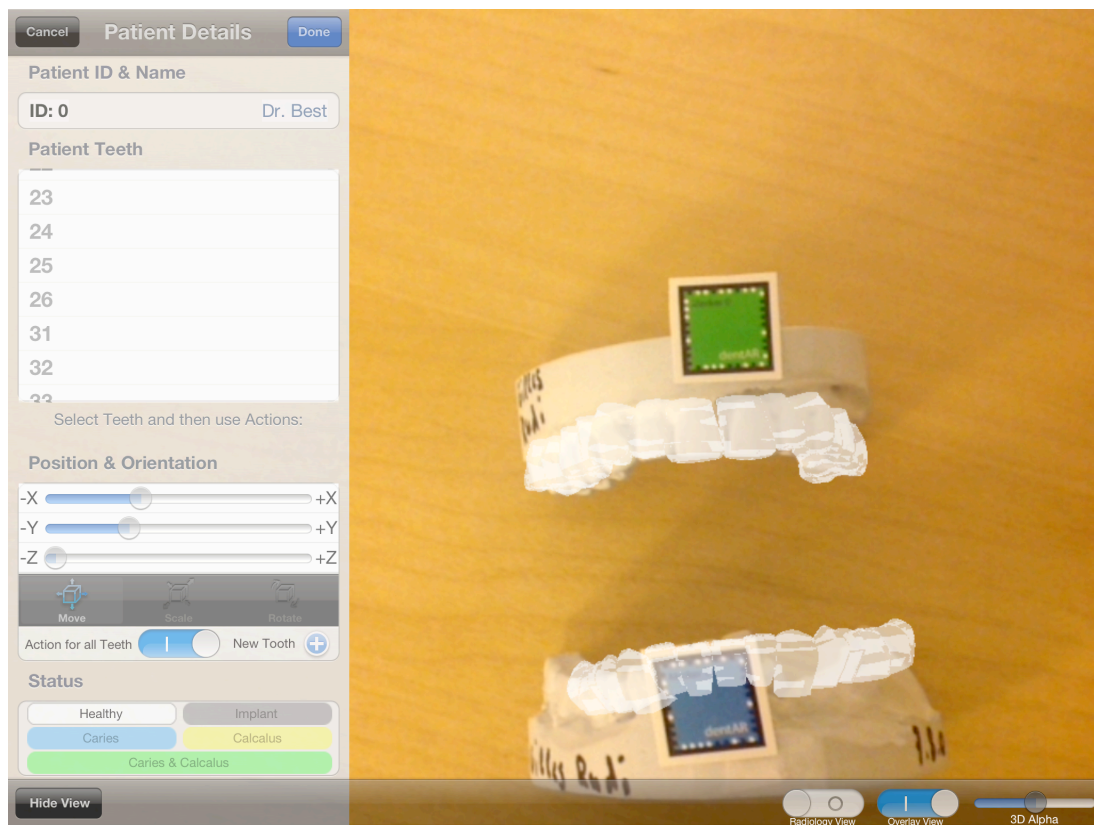
Die Pose-Matrix wird in eine OpenGL konforme 4x4 Matrix umgewandelt. Die nun vorhandene Model-View-Matrix dient als Basis für die weiteren Darstellungen der 3D-Daten. Da mehrere Marker verwendet werden, müssen diese noch unterschieden werden. Anhand des erkannten Markers und dessen umgewandelter Pose-Matrix werden die Zähne der Patienten dargestellt. Die Applikation kann zwischen den Markern für Oberkiefer und Unterkiefer (und dem optionalen Marker für die erweiterten Funktionen) unterscheiden. Wird einer der Marker für Ober- oder Unterkiefer erkannt, werden die virtuellen 3D-Zähne des ausgewählten Patienten sichtbar geschaltet. Die Zahn-Meshes werden dabei anhand ihrer intern initiierten 3D-Positionsdaten passend zum Marker positioniert. Die Anpassung geschieht dabei mit Hilfe der Funktion von *NinevehGL*. Sie skaliert die Zähne und führt die Reposition durch. Weitere Modifikation an den Zähnen

²⁸ Homepage Vuforia „Pose-Matrix-Explained“

werden über den Detail-View-Controller durchgeführt. Die genannten Auffälligkeiten zur Anwendung von Skalierungen und Rotationen wurden bereits in Kapitel 3.2.5 beschrieben. Über die Tabs lassen sich die Slider wählen, die das Verschieben, Skalieren und Rotieren der X-, Y- und Z-Werte realisieren. Zudem wird der Status des Zahns angezeigt und kann zugleich verändert werden. Die dazugehörigen Buttons zeigen die gleiche Farbe, wie die 3D-Zahnfarbe in der AR-View. Die Auswahl der Farben wurde nach Kriterien zur leichten Erkennbarkeit und Unterscheidbarkeit getroffen. Sollten die Patienten-Detailansicht, Overlays oder die überlagerten virtuellen Zähne zuviel Sicht verdecken und die Behandlung stören, können am unteren Bildschirmrand weitere Aktionen ausgeführt werden. Die Ansichten können versteckt, und der Transparenzgrad der 3D-Überlagerung eingestellt werden.



(Bild 13: Simulation)



(Bild 14: AR-Überlagerung auf dem Gipsmodell)

4. Schluss

Im nachfolgenden Kapitel werden zunächst Probleme dargestellt, die während der Arbeit an diesem Projekt entstanden sind. Es wird aufgezeigt, wie mit diesen Problemen umgegangen wurde. Außerdem wird auf Problemsituationen eingegangen die durch den aktuellen Stand der Technik auftreten. Daraufhin werden Verbesserungsverschlage fur das Programm dargestellt. Abschlieend wird das Programm ausgewertet und ein Fazit gegeben.

4.1. Probleme und Verbesserungsvorschlage

Die Programmierprobleme die durch mangelnde Speicherbereichsreservierung oder falsche Zuordnung entstanden sind, wurden durch stetiges uberprufen und Korrigieren des Programmcodes gelost. Die Fehlerbehebung von besonders differenzierten Vorgangen erwies sich als hochst zeitaufwendig. Der genaue Grund konnte durch mangelnde Fehlerindikation in *Xcode* und *Vuforia*s Initiierungsschritten nur schwer gefunden werden.

Neben den Problemen der Programmierung werden nun die Probleme der Bedienbarkeit erlauert. Diese beziehen sich nicht auf Fehler in der Applikation, sondern auf deren Einsatzfahigkeit. Aktuell ist die Steuerung der Applikation durch Touchgesten moglich. Der Arzt muss die Elemente in der Benutzeroberflache aber in der Realitat anders auswahlen konnen. Die Steuerung muss durch die Stimme moglich sein, oder durch eine automatische Selektion durch langeren Blick auf einen Bereich bzw. ein Element auslosbar sein (bei Verwendung eines HMD oder Google Glasses). Da die Applikation zu Zwecken der Simulation entwickelt wurde und die benotigten technischen Moglichkeiten nicht vorhanden sind, ist diese vereinfachte Form, der Steuerung durch Touchgesten gewahlt worden.

Ein weiteres grundlegendes Problem besteht in der Tatsache, dass aktuell noch Marker zum Tracken verwendet werden mussen. Eine vernunftige und stetige Position der Marker am Patienten, die beim nachsten Besuch reproduziert werden kann, ohne die Positionierung korrigieren zu mussen, ist aktuell nicht gewahrleistet.

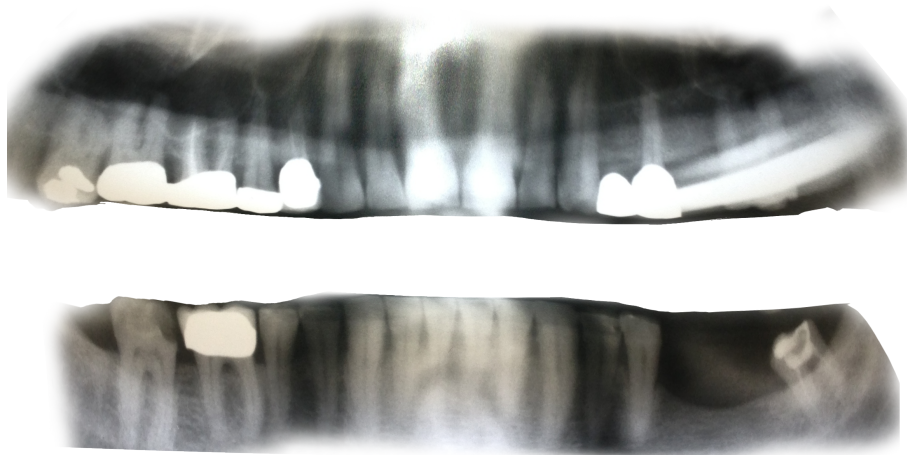
Im Laufe der Implementierung wurden sehr interessante Ideen für sinnvolle Erweiterungsmöglichkeiten der Applikation entwickelt. Optionen zur Anpassung von Farbeinstellungen (Statusfarben), welche eine genauere Einordnung des Zustand des Zahnes eines Patienten (z.B. Grad des Befalls von Karies) mittels einer Differenzierung durch Farbintensität ermöglichen. Durch CT-Daten (Computer-Tomographie-Daten) ließe sich die Idee der Röntgenüberlagerung auch in das 3D-Zahnmodell integrieren. Die exakte Position von Karies oder sonstigen Auffälligkeiten am Zahn könnte so ebenfalls in das virtuelle 3D-Zahnmodell integriert werden. Eine andere Idee befasst sich damit, dass die Röntgenbilder unterteilt werden und jeder Zahn seine eigene Röntgenbild-Textur erhält, die bei Bedarf auf den Zahn gelegt werden kann. Die Integrierung einer Zeitleiste, bei der die Werte jeder Behandlung abgefragt werden können, würde eine Dokumentation der Patientenbehandlungen ermöglichen. Die Darstellung des Heilungsverlaufes kann so in einzelnen Schritten (oder als Animation) angezeigt werden. Ein Behandlungsverlauf ist auch im Bereich der Kieferorthopädie von Nutzen, bei dem die Verschiebung der Zähne durch eine Spange gezeigt werden kann.

All diese Möglichkeiten sind natürlich nur dann relevant, wenn die Problematik der Position der Marker gelöst wird (beispielsweise durch eine konkretes Zahnarzt-Equipment, welches zur fixen Befestigung der Marker verwendet wird oder ein markerloses Tracking). Ein Alternative ist in der aktuellen Version der *Vuforia SDK*²⁹ möglich. Statt der vorher selbst gewählten und definierten Image Marker bzw. der vorgegebenen Frame Marker, können die Image Marker „on the fly“ erstellt werden. Dabei wird mit der Kamera ein Bild gemacht und dieses dient dann als Grundlage zum Tracken. Bezogen auf die Applikation und die Zahnmedizin könnte so jeweils ein Bild vom Oberkiefer (bzw. oberen Teil des Gesichts) und Unterkiefer (bzw. unterem Teil des Gesicht) gemacht werden. Diese patientenbezogenen Bilder können dann als Marker verwendet werden. So entsteht der Eindruck einer ‚markerlosen‘ Vorgehensweise. Zu Beginn der Arbeit war diese Funktion jedoch noch nicht verfügbar und konnte bei der Implementierung nicht berücksichtigt werden.

²⁹ Für diese Arbeit wurde *Vuforia SDK* nicht verwendet, da es erst während der Implementation veröffentlicht wurde.

Sonstige Schwierigkeiten wurden bereits im Hauptteil an den jeweils relevanten Stellen beschrieben.

Die optionale Möglichkeit, mittels eines dritten Markers eine Röntgenbildüberlagerung zu ermöglichen, wurde leider nicht zu 100% erfüllt. Die Grundfunktionen dafür wurden gelegt, konnten aber durch Zeitmangel nicht komplett realisiert werden. Zur Vorbereitung der Funktion wurde ein vorhandenes Röntgenbild in Ober- und Unterkiefer getrennt und passend beschnitten. Damit die Überlagerung einigermaßen deckungsgleich stattfindet wurde das Bild leicht verzerrt und „begradigt“. Weiterführende Funktionen im Programm, um das Bild als eine Textur auf ein Mesh zu legen und bei Erkennung des passenden Markers, dieses als Teilausschnitt oder ganz darzustellen, fehlen noch.



(Bild 15: Vorbereitete Texturen auf dem Röntgenbild)

4.2 Auswertung

Das Ziel dieser praktischen Arbeit, eine funktionierende Applikation zu erarbeiten, wurde erfüllt. Dargestellt wird ein simulierter Prototyp, der aufzeigt, dass AR für einen behandelnden Zahnarzt machbar ist und ihn bei seiner Arbeit unterstützen kann. Die Frage, ob die Arbeit des Zahnarztes im Jahre 2013 dadurch erleichtert wird, muss jedoch verneint werden. Die genannten Probleme bei der Stetigkeit der Marker-Position, um wiederkehrende Behandlungen zu ermöglichen, sind ausschlaggebend für eine aktuelle Untauglichkeit. Wird das Problem gelöst, kann der Einsatz einer überarbeiteten Applikation und die Überlagerung von Informationen, wie 3D-Modelle, den Arzt unterstützen. Sollten diese den Arzt dann doch einmal stören oder behindern, kann man die Anzeige anpassen oder ausblenden. Die Applikation bezieht sich auf eine Simulation einer Anwendung in der Zukunft. Die aktuelle Problemfälle lassen sich durch die technische Weiterentwicklung und den Fortschritt in der Zukunft lösen. Daher kann festgestellt werden, dass die hypothetische Fragestellung dieser Bachelor-Thesis - wie, ob und was dem Zahnarzt während der Behandlung helfen kann - erfolgreich beantwortet wurde.

4.3 Ausblick und Fazit

Generell zeigte die Durchführung der Arbeit, wie in der Einleitung beschrieben, dass im Bereich der Augmented Reality tatsächlich ein enormes Potential steckt. Jeder der die Applikation gesehen hat, war trotz ihres eingeschränkten Funktionsumfangs und ihrer leider fehlenden Alltagstauglichkeit (da sie nur eine hypothetische Arbeit für einen behandelnden Arzt in der Zukunft ist) erstaunt und begeistert. Basierend auf der Weiterentwicklung in der Marker-Forschung, sowie in Forschungen im Bereich der Interaktion und Selektion mit Hilfe von Blickkontakt und Stimme oder der Erkennung von Muskelbewegungen, wird ein zukünftiger Einsatz gewährleistet. Die Verbesserungsvorschläge und möglichen Zukunftseinsätze (neben den Integrierten) zeigen sinnvolle Erweiterungsmöglichkeiten.

Ziel war es auch die eigenen Fähigkeiten auszutesten und eine funktionierende iOS App von Anfang bis Ende vollkommen eigenständig zu entwickeln. Zudem sollten die bisher nur in der Theorie erfahrenen Möglichkeiten der Augmented Reality durch einen Anwendungsfall in der Praxis verdeutlicht werden. Die Hürden während der Entwicklung und der Implementierung wurden, trotz der Verzögerungen durch die Fehlermeldungen und daraus stockender Entwicklung, im Endeffekt überwunden. Obgleich einer Verschiebung im Zeitplan und spontan benötigter Flexibilität beim Programmieren, konnte viel Wissen gefestigt und das Verständnis für die AR und die eigenen Programmierfähigkeiten erweitert werden. Das fertige Produkt erfüllt mich umso mehr mit Stolz, wenn ich die überwundenen Probleme betrachte. Auch in Zukunft möchte ich mich intensiv mit dem Themengebiet der AR beschäftigen und dieses Projekt weiterentwickeln.

Bilderverzeichnis

Bild 1: Logo der entwickelten Applikation	Deckblatt
Bild 2: Google Glass	S.8
(http://www.google.com/glass/start/what-it-does/	
(Abrufdatum: 18.03.2013))	
Bild 3: 3D- Zahnmodell von VikingDigital	S.11
Bild 4: GipszahngGebiss mit Markern	S.13
Bild 5: Selbst erstellte Test Image Marker	S.14
Bild 6: Original- <i>Vuforia</i> Frame Marker	S.15
Bild 7: Selbst erstellte Frame Marker	S.15
Bild 8: Programmstruktur & Datenmodellierung	S.17
Bild 9: Storyboard	S.19
Bild 10: Benutzeroberfläche der Patientenliste	S.23
Bild 11: Benutzeroberfläche der Patientendetails	S.26
Bild 12: Koordinatensystem der Pose-Matrix	S.27
 (https://developer.vuforia.com/resources/dev-guide/pose-matrix-explained	
(Abrufdatum: 18.03.2013))	
Bild 13: Simluation: AR-Überlagerung auf dem Gipsmodell	S.28
Bild 14: AR-Überlagerung auf dem Gipsmodell	S.29
Bild 15: Vorbereitete Texturen auf dem Röntgenbild	S.32

Quellenverzeichnis

Literaturquellen:

Gamma, Erich & Helm, Richard & Johnson, Ralph & Vlissides, John (1994). Design Patterns. Elements of Reusable Object-Oriented Software. Amsterdam: Addison-Wesley Longman

Mehler-Bicher, Anett & Reiß, Michael & Steiger, Lothar (2011). Augmented Reality, Theorie und Praxis. München: Oldenburg Wissenschaftsverlag GmbH

Milgram, Paul & H. Takemura, A. & Utsumi, F. Kishino (1994). Augmented Reality: A class of displays on the reality-virtuality continuum. SPIE Vol. 2351. Telemicroscopy and Telepresence Technologies, Bellingham: WA.

Internetquellen:

Amour, Isaam (18.03.2013): „Samsung: Smartphones mit flexiblen Displays in ferner Zukunft“ unter <http://www.pc-max.de/news/smartphones/samsung-smartphones-mit-flexiblen-displays-in-ferner-zukunft> (Abrufdatum: 18.03.2013)

Behrens, Christoph (10.12.12): „Smartphone-Boom katapultiert Samsung an die Spitze“ unter http://www.focus.de/finanzen/news/unternehmen/samsung-haengt-apple-ab-smartphone-boom-katapultiert-samsung-an-die-spitze_aid_878871.html (Abrufdatum: 18.03.2013)

Bodonge, Andreas (09.03.2013): „Google: Smartphone mit Back-Panel- Touchscreen“ unter <http://www.hardware-infos.com/news/4312/google-smartphone-mit-back-panel-touchscreen.html> (Abrufdatum: 18.03.2013)

Homepage von Apple Xcode:

<https://developer.apple.com/technologies/tools/> (Abrufdatum: 18.03.2013)

Homepage der Google Glasses:

<http://www.google.com/glass/start/> (Abrufdatum: 18.03.2013)

Homepage von Gartner:

<http://www.gartner.com/technology/home.jsp> (Abrufdatum: 18.03.2013)

Homepage von NinvehGL:

<http://nineveh.gl/> (Abrufdatum: 18.03.2013)

Homepage von NinvehGL Reference:

<http://nineveh.gl/docs/index.html> (Abrufdatum: 18.03.2013)

Homepage von Maya:

<http://usa.autodesk.com/maya/> (Abrufdatum: 18.03.2013)

Homepage der CAMPAR-Abteilung der TU München: (Computer Aided Medical Procedures & Augmented Reality) Forschung und Entwicklung von medizinischen AR Anwendungen: <http://campar.in.tum.de/Chair/ResearchIssueMedAR>

Projekt-Narvis: http://campar.in.tum.de/view/Chair/NarvisLab#CAMP_Research_Projects_NARVIS#CAMP_Research_Projects_NARVIS

(Abrufdatum: 18.03.2013)

Homepage von Turbosquid: Viking Digital Gums & Tongue 3D Model

<http://www.turbosquid.com/3d-models/gums-tongue-obj/650105>

(Abrufdatum: 18.03.2013)

Homepage von Qualcomm Vuforia:

<http://www.qualcomm.com/solutions/augmented-reality> (Abrufdatum: 18.03.13)

Homepage von Vuforia Developer

<https://developer.vuforia.com> (Abrufdatum: 18.03.13)

Homepage von Vuforia „Pose-Matrix-Explained“

<https://developer.vuforia.com/resources/dev-guide/pose-matrix-explained> (Abrufdatum: 18.03.13)

Rampacher, Carsten (14.03.2012): „TOPS UND FLOPS: Aktuelle Entwicklung beim Smartphones“ unter http://www.aredvd.de/hardware/2012/smartphone_top_flop.shtml (Abrufdatum: 18.03.2013)

Steimels, Dennis (01.06.2013): „Wie alles begann: Die Geschichte der Smartphones.“ unter <http://www.pcwelt.de/ratgeber/Handy-Historie-Wie-alles-begann-Die-Geschichte-des-Smartphones-5882848.html> (Abrufdatum: 18.03.2013)

„Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-ROM). Ich stimme einer Veröffentlichung in der Bibliothek und im Internet zu.“

Ort, Datum

Unterschrift