Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht

Arazm Hosieny

Pervasive Gaming Framework

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Arazm Hosieny

Pervasive Gaming Framework

Seminarausarbeitung im Rahmen der Veranstaltung Anwendungen 1
im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Olaf Zukunft

Abgegeben am 08. März 2009

Meinen Nachfahren,
meiner Frau
und meinem Vater gewidmet

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

Pervasive Gaming ist eine nahtlose Integration von Spielen in unsere physische Welt. Mit Hilfe von mobilen Geräten und deren spezifischen Eigenschaften soll eine neue Art von Spielerlebnis geschaffen werden. Das wesentliche Ziel ist es, eine Brücke zwischen realer und virtueller Welt zu schlagen [Carsten Magerkurth 2009]. Auch diese Arbeit mit dem Schwerpunkt Pervasive Gaming Framework kann dem genannten Themengebiet zugeordnet werden. Auf den folgenden Seiten werden die Motivation und die Zielsetzung dieser Arbeit erörtert. Anschließend folgen Informationen über den inhaltlichen Aufbau der Arbeit.

Die Ausarbeitung ist im Rahmen des Masterprojekts der HAW-Hamburg entstanden und konzentriert sich auf das Pervasive Gaming. Zu Beginn des Projekts wurde der Fokus auf ein Framework für mobile Spiele und zwei drauf basierende Anwendungen gelegt. Das im Wintersemester 2008/2009 durchgeführte Projekt bestand aus den folgenden Mitgliedern: Thomas Preisler, Sascha Kluth, Dennis Dedaj, Tobias Huntzler, Peter Salchow, Arazm Hosieny, Amine El Ayadi und Julia Pliszka. Das Projekt wurde von Prof. Dr. Olaf Zukunft betreut.

1.1 Motivation und Zielsetzung

Digitale Spiele sind in verschiedenen Bereichen des Alltags und der Unterhaltung verwurzelt. Sie wurden im Verlauf der Zeit immer mehr perfektioniert, allerdings konzentrierte man sich meistens auf bekannte Spielarten, wie zum Beispiel Sportspiele und Strategiespiele. Dabei wurde beim Konzeptentwurf von Spielen der soziale Kontakt der Spieler sehr oft vernachlässigt. Damit der Spieler nicht von der äußeren Welt isoliert wird, entstand die Idee, das Spiel in das reale Leben zu integrieren. Ermöglicht wird dies durch den weiter anhaltenden Fortschritt der Informationstechnik. Viele Geräte in Kombination mit den Software-Produkten bieten ortsbezogene Dienste [Georg Treu u. Küpper 2009]. Die Entscheidung aller Projektmitglieder, sich mit ortsbezogenen Spielen und dem Pervasive Gaming Framework zu beschäftigen, kam daher, weil alle großes Interesse an Location based Services, sowie ein allgemeines Interesse an mobilen Systemen und Spielen haben.

Das Ziel des Projekts war die Entwicklung eines Frameworks für mobile Anwendungen. Die mobilen Anwendungen sind im Bereich des „Pervasive Gaming“ angesiedelt. Das Projekt

wurde in zwei Gruppen (Anwendung und Framework) aufgeteilt, wobei es eine weitere Aufteilung innerhalb der Anwendungsgruppe gab. Die Frameworkgruppe bestand aus Tobias Huntzler, Peter Salchow und Arazm Hosieny, die Anwendungsgruppe I aus Julia Pliszka und Amine El Ayadi, die Anwendungsgruppe II aus Thomas Preisler und Dennis Dedaj bestehend. Während die Frameworkgruppe ein Framework für pervasive Spiele und Anwendungen entwicklt hat, beschäftigte sich die Anwendungsgruppe II mit der Entwicklung eines mobilen Rollenspiels „The Word Within“. Die Anwendungsgrupppe I war mit der Architektur und Realisierung einer Anwendung mit dem Namen „iLife“ beschäftigt. Zur Vertiefung der Thematik dieser Ausarbeitung sei der Leser auf die jeweiligen Projektberichte der Gruppen verwiesen. Im Vordergrund stand, die entwickelten Anwendungen in Android, einer Plattform für mobile Geräte, zu integrieren. Für Android spricht die Entwicklungsmöglichkeit in der favorisierten Programmiersprache Java. Dabei sollten praktische Erfahrungen gesammelt und Kenntnisse in mobilen Technologien erlangt werden.

Im folgenden Kapitel 2 werden die Anforderungen an die Entwicklung des Pervasive Gaming Framework herausgearbeitet. Die Rolle des Verfassers dieser Ausarbeitung ist in der Frameworkherstellung definiert. Daher wird die Analyse der vorliegenden Ausarbeitung das Hauptaugenmerk auf diese Teilbereichen des Projekts und dem Verfasser zugeordneten Tätigkeiten legen.

1.2 Aufbau der Arbeit

Zum Aufbau: Abschnitt 2 schneidet die Anforderungen an das Pervasive Gaming Framework. In Abschnitt 3 werden Realisierungskonzepte vorgestellt, die man für die Entwicklung des Frameworks einsetzen könnte. Abschnitt 4 zeigt Vertiefungspunkte in einigen Bereichen der Architektur. Die Ausarbeitung schließt in Abschnitt 5 mit einem Fazit ab und gibt einen kurzen Ausblick.

2 Anforderungen

Bevor in diesem Kapitel Anforderungen an das Framework gestellt wurden, waren sich alle Projektteilnehmer darüber einig, dass das Framework als eine Sammlung von Services zu sehen ist, aus der der Entwickler die für ihn relevanten Module separat verwenden kann. Die Anforderungen ergeben sich aus den zu entwickelnden Anwendungen (Rollenspiel und iLife). Diese wurden im Gespräch mit den einzelnen Anwendungsgruppen erarbeitet. Dabei sollen zuerst die einzelnen Anwendungen separat betrachtet und, als nächstes die Abstraktion der beiden erarbeitet werden.

2.0.1 Anforderungen an die Architektur durch iLife

Gespräche mit dem iLife-Team [EL-AYADI 2009] [PLISZKA 2009] haben ergeben, dass die Anforderungen sehr anwendungsspezifisch sind, und eine Implementierung seitens des Frameworkteams nicht sinnvoll wäre. Es würde mehr Sinn machen, die Funktionen in Modulen zu kapseln und die Verwaltung der Module dem Framework zu überlassen. Die Anwendung iLife beinhaltet nämlich verschiedene Daten (z.B. Videos, Bilder, Ereignisse, Texte), sie werden im Internet in Blogs veröffentlicht. Es existieren Schnittstellen der Internetanwendungen für die Bedienungen. Das Framework müsste diese speziellen Schnittstellen implementieren. Eine weitere Herausforderung für das Framework stellte die Positionsbestimmung innerhalb von Räumen dar. Daten, die auf dem Endgerät erstellt werden, würden mit Geodaten versehen, um zeitliche und räumliche Abfolgen zu rekonstruieren. Mittels GPS ist die Bestimmung der Positionen innerhalb von Gebäuden nicht möglich. Aus diesem Grund müssten andere Alternativen gesucht werden. Zudem ermöglicht iLife Benutzern, aktuelle Ereignisse untereinander auszutauschen. Sobald ein Benutzer einen Eintrag in seinen Kalender macht, werden interessierte Benutzer darüber informiert. Falls ein Benutzer nicht erreichbar ist, bekommt er asynchron die Nachricht später. Das Framework muss in der Lage sein Daten an einer zentralen Stelle zu persistieren und den Austausch von Informationen zu ermöglichen. Diese Funktionen werden daher von der Anwendung bereitgestellt.

2.0.2 Anforderungen an die Architektur durch das Rollenspiel

An dieser Stelle sei genannt, dass die folgenden Anforderungen sich auf das Framework beziehen. Für Anforderungen an das Rollenspiel sei auf die Ausarbeitung von Dedaj u. Preisler verwiesen [DEDAJ 2009].

Avatare der Spieler müssen gespeichert und überall zugreifbar sein. Durch diese Anforderung muss das Framework zwei Dienste zur Verfügung stellen: zum einen die Datenpersistenz (Persistenz der Avatare) und zum anderen einen einen zentralen Server, der den Zugriff auf Avatare gewährleistet. Daraus resultiert, dass die Endgeräte und der Server untereinander Daten austauschen müssen.

Die Realisierung eines Duells zwischen zwei Spielern ist eine weitere Anforderung. Ein Spieler wählt einen anderen Spieler auf dem Endgerät aus, um ihn anzugreifen. Der angegriffene Spieler wird darüber benachrichtigt. In Abhängigkeit davon, wie viel Erfahrung und Stärke die Avatare besitzen, wird ein Schaden zugefügt. Diese Anforderung an das Framework, erfordert eine Kommunikation zwischen den Endgeräten. Der Angriff ist eine spezifische Funktion des Rollenspiels. Dazu wäre eine Abstraktion erforderlich, sodass alle möglichen Daten kommuniziert werden. Die Spielform des Duells erfordert außerdem, dass die Spieller zur gleichen Zeit eine bestimmte Entfernung zueinander einhalten. Die Avatare werden dazu auf der realen Weltkarte abgebildet. Auch hierbei ist die Positionsbestimmung seitens des Frameworks notwendig. Zusätzlich müssen die in einem bestimmten Umkreis stehenden Mitspieler ermittelt werden können.

Mittels eines Wii-Controllers sollen bestimmte Bewegungen und Gesten des Spielers erkannt werden. Diese Bewegungen sollen dann bestimmte Aktionen im Spiel auslösen. Für diese Anforderung muss das Framework die Daten der Hardware registrieren und zur Lösung von Aktionen umsetzen können. Welche Aktionen einzelnen Bewegungen zugeordnet sind, wird von der Anwendung dem Framework mitgeteilt.

2.0.3 Zusammenfassung aller Anforderungen

Im vorigen Abschnitt sind wichtige Details über die Anwendungen für die Konzeption des zu entwickelnden Frameworks beschrieben und die einzelnen Entscheidungen, die hierbei eine Rolle spielen, diskutiert worden.

Dieser Abschnitt fasst alle Anforderungen mit den umzusetzenden Komponenten und deren Zusammenspiel zusammen. Anschließend wird in Kapitel 4 beschrieben, wie das Framework praktisch umgesetzt worden ist, und welche Untersuchungen für die Umsetzung der Anforderungen notwendig waren, worauf der Verfasser speziell seinen Schwerpunkt gelegt hat.

Die Auswahl der Komponenten ist so zu treffen, dass sie jeder Zeit ausgetauscht oder erweitert werden können. Im Folgenden werden die benötigten Komponenten aufgelistet:

Kommunikationskomponente:

Endgeräte müssen untereinander Objekte austauschen können (Endgerät-Endgerät). Genauso muss es möglich sein, Objekte zwischen Endgerät und Server austauschen zu können (Endgerät-Server). Der Verfasser hat sich in diesem Fall mit Web Services beschäftigt. Näheres dazu findet sich im nächsten Kapitel.

Persistenzschicht:

Daten müssen sowohl auf dem Endgerät, als auch auf dem Server persistiert werden können. Für die Persistenz der Objekte auf dem Client ist die Datenbank (SQLite) erforderlich. Für den Server kommen die Open Source Datenbanken MySQL, PostgreSQL oder HSQL in Frage.

Positionskomponente: Teilnehmer der Anwendungen müssen in der Lage sein, die eigenen und die Positionsdaten der anderen Teilnehmer ermitteln können

Bewegungserkennungskomponente:

Das Gerät muss Bewegungsabläufe erkennen und auswerten können. Bestimmte Bewegungnen müssen Aktionen auslösen können.

3 Realisierung

Um eine gemeinsame Vorstellung über das Gesamt-System zu bekommen, wurden mehrere Architekturentwürfe im Framework-Team konzipiert. Zwei Entwürfe haben sich am Ende durchgesetzt, wobei das Framwork 3.2 der Sieger war, da es auf das Framework von Android aufsetzt und die Funktionen von Android kapselt. In einem anschließenden Gruppenmeeting haben sich alle Gruppenmitglieder auf diesen Entwurf geeinigt, der als Grundlage für das Projekt dienen sollte.

3.1 Realisierung der Architektur

Wie bereits erwähnt standen sich am Ende zwei Architekturen gegenüber. Die erste Architektur 3.1 setzte direkt auf den Android-Stack. Module und Anwendungen dienten als Erweiterung des Frameworks. Jedoch müsste das Framework das gesamte Modul- und Anwendungsmanagement bewältigen. Auf Grund unterschiedlicher Anforderungen, wäre dessen Einsatz denkbar gewesen, jedoch bot die zweite Architektur 3.2 eine feingranulare Verteilung der Anwendungen und Module. Zudem wurde eine Erweiterung um eine technische Schicht, Eventmanagement (Überwachung der ausgelösten Ereignisse durch Module), Dependency-Manager (Abhängigkeitsmanagement der Modulen untereinander) und Modulmanagement (Laden von Modulen) vorgenommen.

Die oben beschriebene Architektur 3.2 wäre eine ideale Lösung. Doch leider ist das Framework-Team aus zeitlichen Gründen von den beiden Architekturen abgekommen. Die Anwendungsentwickler konnten ihre Arbeit nicht fortsetzen, solange das Framework-Team diese Services zur Verfügung gestellt hatte. Am Ende hat man sich dafür entschieden, die Services des Frameworks als Android-Anwendungen zu starten. Android übernimmt die Überwachung dieser Services. An dieser Stelle sei zur Vertiefung auf die Projektberichte der anderen Frameworkmitglieder [HUTZLER 2009] [Salchow 2009] verwiesen.

Im nächsten Kapitel werden die relevanten Services, die für die Realisierung der Anforderungen notwendig waren vorgestellt.

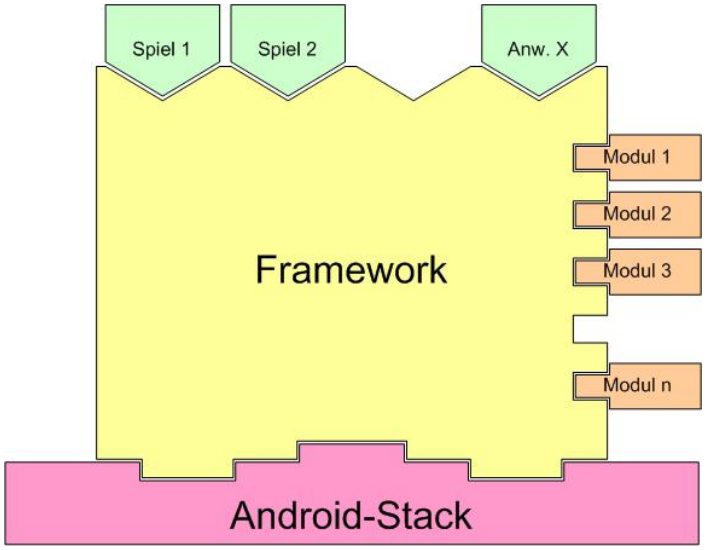

Abbildung 3.1: Pervasive Gaming Framework - Architektur I

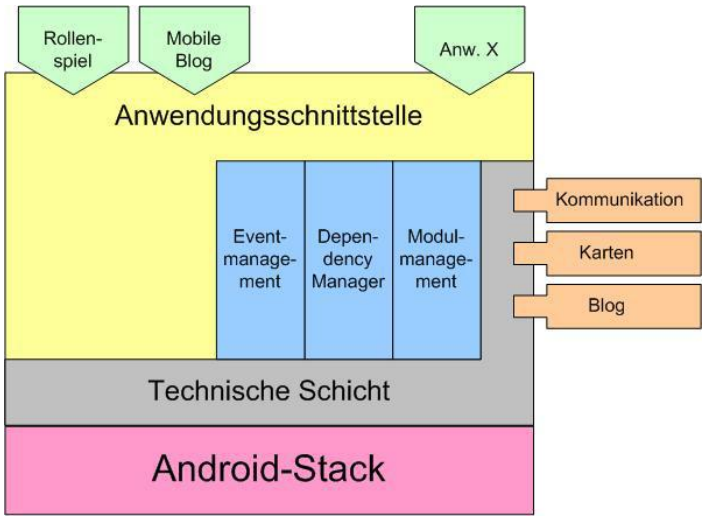

Abbildung 3.2: Pervasive Gaming Framework - Architektur II

4 Angewendete Technologien und weitere Tätigkeiten im Projekt

Im folgenden Kapitel werden die Rahmenbedingungen für das Projekt erläutert. Anschießend werden Technologien betrachtet die für den Einsatz des Frameworks notwendig waren.

4.1 Hardware und Software

Vor dem Projektbeginn standen drei Endgeräte zur Auswahl: Android, IPhone und Windows Mobile. Trotz der kritischen Situation bezüglich des Lieferdatums des Androids, wurde Android von allen Projektteilnehmern favorisiert. Das KO-Krierium für iPhone war die erforderliche Einarbeitung in die unbekannte Programmiersprache Objective C und hohe Anschaffungskosten. Die Gefahr bei der Einarbeitung in die Programmiersprache bestand darin, von der wichtigen konzeptionellen Arbeit abgelenkt zu werden. Windows Mobile wurde ebenfalls auf Grund der Programmiersprache C# abgelehnt. Da alle Teilnehmer der Programmiersprache Java mächtig sind und den Schwerpunkt auf konzeptionelle Kriterien legen, wurde einstimmig für die Verwendung von Android gestimmt. Ein Risiko bestand weiterhin, dass die Anwendungen in der Projektzeit nur im Emulator getestet werden können. Trotzdem war er der Sieger. Der Verfasser dieser Ausarbeitung war unter anderem für die Beschaffung des Android-Google-Phone zuständig.

Android ist ein Software-Stack für mobile Geräte, das aus Betriebssystem, Middleware und Anwendungen besteht. Android bildet die Abstraktionsschicht zur darunter liegenden Hardware. Hauptsächlich wurde Android in Java entwickelt und erlaubt den Softwareentwicklern die Programmiersprache Java zu verwenden [GOOGLE 2008].

4.2 Kommunikationskomponente mittels Webservices

Das in diesem Projekt entwickelte Framework besteht aus einer Server- und einer Clientkomponente 2.0.3. Die Kommunikation dieser Komponenten sollte über Web Services realisiert werden.

Aus diesem Grund werden in diesem Abschnitt die Grundlagen von Web Services kurz erläutert. Web Services sind einige Standards, über die die Kommunikation zwischen Endgeräten ermöglicht wird. Die wichtigsten Standards sind SOAP, WSDL und UDDI. Der SOAP dient der Übertragung von Nachrichten. Er legt den Aufbau von Nachrichten fest. WSDL (Web Service Description Language) beschreibt die Web Services. Das WSDL-Dokument ist die Schnittstelle, über die Web Services angesprochen werden. Universal Description, Discovery and Integration of Web Services (UDDI) ist ein Web Service-Verzeichnisdienst. Web Services-Anbieter können ihre Dienste in UDDI-Verzeichnissen veröffentlichen. Web Services-Nutzer können auf diese Verzeichnisse zugreifen [HAUSER 2004].

Für die Lösung der Kommunikationskomponente war vorgesehen, Web Services zu nutzen. Dafür kam JAX-WS 2.0, ein Bestandteil von Java 1.6 API, in Frage. Bei der prototypischen Implementierung wurde vom Verfasser festgestellt, dass Android keine Webservices unterstützt. Daher mussten vom Framework-Team Alternativlösungen mittels TCP/IP gefunden werden.

4.3 Persistenzkomponente des Servers

Wie bereits im Kapitel 2.0.3 ausgeführt, lautete die Anforderung, eine Persistenzkomponente auf der Client und eine auf dem zentralen Server zu integrieren. Auf der Serverseite kam MySQL als relationale Datenbank und zum Persistieren der Objekte Hibernate zum Einsatz. Es wird als nächstes kurz erläutert, was unter Persistenz verstanden wird, und welche Argumente für die Verwendung von Hibernate sprechen.

Persistenz

Fast alle Anwendungen erfordern persistente Daten. „Langlebigkeit der Daten“ ist eines der grundlegenden Konzepte in Anwendung und Entwicklung. Wenn ein Informationssystem die Daten der Benutzer nicht erhalten würde, wenn der Host abgeschaltet wurde, wäre dieses System vom geringem praktischem Nutzen. In unserem Fall wäre dies das PDA. Wenn man über Persistenz in Java spricht, meint man in der Regel die Speicherung von Daten in einer relationalen Datenbank mit Hilfe von SQL. Man beginnt mit einem kurzen Blick auf die Technik und wie man es in Java einbinden kann. Auf Grundlage dieser Informationen kann man

weiter überlegen, wie es in objektorientierte Anwendungen umgesetzt wird. Relationale Datenbanken ist eine bekannte Technologie, um Daten zu verwalten. Sie bietet ein sehr flexibles und robustes Konzept für das Daten-Management. Ein solides Verständnis des relationalen Modells und SQL ist eine Voraussetzung, um Hibernate effektiv anzuwenden. Man muss die eigenen Kenntnisse von SQL nutzen, um die Leistung der Anwendung zu steigern. Hibernate automatisiert viele sich wiederholende Aufgaben, aber die eigenen Kenntnisse über die Persistenz-Technologie müssen darüber hinausragen, wenn man die volle Kraft der modernen SQL-Datenbanken nutzen möchte. Hibernate-Anwendungen definieren persistente Klassen in den Datenbanktabellen [Bauer u. King 2004] [Bauer u. King 2006] [Parick Peak 2005].

4.4 Server

An Hand der Anforderungen wurde im Verlauf des Projekts ein Administrator bestimmt, der für die Wahl des Server-Betriebssystems zuständig war und die Betreuung des Servers über den gesamten Zeitraum übernahm. Da der zur Verfügung gestellte Server, ein sogenannter VMware ESX Server war, war es erforderlich, sich in diese Thematik einzuarbeiten. Als Server-Betriebssystem wurde auf Grund bereits vorhandener Kenntnisse spontan ein Windows Server 2003 bestimmt. Dieses Betriebssystem sollte als virtuelle Maschine im ESX Server laufen. Ein kurzes Beispiel verdeutlicht dieses Szenario:

Ein Host (Wirt), ist ein realer Rechner, auf dem die virtuellen Maschinen laufen. Auf dem Host kann irgend ein beliebiges System installiert werden, unter dem anschließend eine Virtualisierungssoftware installiert wird. Dieser Host ist entweder ein Desktop-PC oder ein Server mit gewöhnlicher Arbeitsoberfläche, auf dem die virtuellen Anwendungen laufen können. Nach diesem Prinzip arbeiten VMware Server, VMware Worksatationen und der Player von Microsoft Virtual PC/Server.

Im Gegensatz dazu läuft wie in der Abbildung 4.1 zu erkennen ist, der VMware ESX Server direkt auf der Hardware und benötigt kein Wirtsbetriebssystem. Mehr zu diesem Thema kann dieser Quelle entnommen werden [Ahnert 2008].

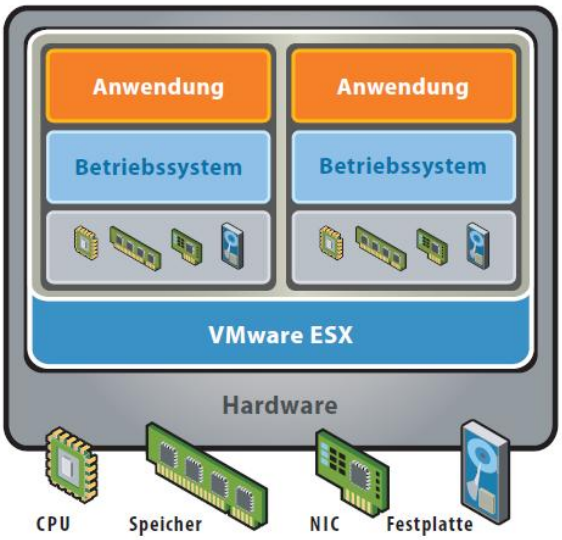

Abbildung 4.1: VMware ESX - Server
[vmware.com 2009]

5 Fazit

Die Projektmitglieder haben unterschiedliche Interessenschwerpunkte gehabt. Durch die frühzeitige Einteilung in unterschiedliche Aufgabenbereiche, Anwendung und Framework, konnte sich jeder ambitioniert in sein jeweiliges Interessengebiet vertiefen. Die wöchentlichen Meetings haben dazu beigetragen, den Stand der Entwicklung zu messen und sich ein Feedback vom gesamten Team zu holen, auch wenn es im Laufe der Planungsphase zu intensiven Diskussionen kam. Das Projektmanagement hat im Verlauf des Projekts mit allen Mitgliedern Risiken analysiert und versucht dagegen zu steuern. Dennoch kam es zu Verzögerungen im Projekt, da man auf Lieferungen warten musste. Sehr positiv war die Erfahrung, neue Techniken auszuprobieren und zu erlernen, wobei das Einarbeiten zu einer Entwicklungsverlängerung geführt hat. Das hatte wiederum zur Folge, dass die eingeplanten Testphasen nicht realisiert werden konnten.

Die Präsentation am Ende des Projekts zeigte, dass die einzelnen Ergebnisse insgesamt jedoch von Erfolg gekrönt waren. Beide Anwendungsgruppen (iLife und Rollenspiel) haben ein positives Resultat gezeigt. Das Rollenspielteam hat von den Funktionen des Frameworks eher profitiert, da es enger mit dem Framework-Team gearbeitet hat, während iLife unabhängig vom Framework entwickelt wurde.

5.1 Ausblick

Für die Nachfolger des Masterstudiengangs stehen alle Geräte komplett bereit. Im Projekt wurde ein Pervasive Gaming Framework entwickelt. In diesem Zusammenhang besteht die Möglichkeit, darauf aufzubauen und die hier gewonnen Erfahrungen zu berücksichtigen und eventuell die Architektur 3.2 als Grundlage zu verwenden oder erweitern.

Literaturverzeichnis

Ahnert 2008
AHNERT, Sven: *Virtuelle Maschinen mit VMware und Mircosoft*. 2. Aufl. 2008

Bauer u. King 2004
BAUER, Christian ; KING, Gavin: *Hibernate in Action*. Oktober 2004

Bauer u. King 2006
BAUER, Christian ; KING, Gavin: *Java Persistence with Hibernate*. Second Edition of Hibernate in Action. November 2006

Carsten Magerkurth 2009
CARSTEN MAGERKURTH, Dan G. Timo Engelke E. Timo Engelke: *A Component Based Architecture for Distributed, Pervasive Gaming Applications*. `http://portal.acm.org/results.cfm?coll=Portal&dl=Portal&CFID=24365142&CFTOKEN=60152450`. Version: Februar 2009

DEDAJ 2009
DEDAJ, Thomas Dennis ; P. Dennis ; PREISLER: *Pervasive Gaming Framework*. 2009

Diep Dao u. Wang 2002
DIEP DAO, Chris R. ; WANG, Jinling: *Location-based services: technical and business issues*. December 2002

EL-AYADI 2009
EL-AYADI, Amine: *Pervasive Gaming Framework*. 2009

Georg Treu u. Küpper 2009
GEORG TREU, Peter R. ; KÜPPER, Axel: *Efficient Indoor Proximity and Separation Detection for Location Fingerprinting*. `http://portal.acm.org/results.cfm?coll=GUIDE&dl=ACM&CFID=23236085&CFTOKEN=73026962`. Version: Februar 2009

GOOGLE 2008
GOOGLE: *Android - An Open Handset Alliance Project*. `http://code.google.com/android/`. Version: Juli 2008

HAUSER 2004
HAUSER, Ulrich M. Tobias ; LÖWER L. Tobias ; LÖWER: *Web Services - Die Standards*. 2004

HUTZLER 2009
HUTZLER, Tobias: *Pervasive Gaming Framework*. 2009

KLUTH 2009
KLUTH, Sascha: *Pervasive Gaming Framework*. 2009

Parick Peak 2005
PARICK PEAK, Nick H.: *Hibernate Quickly*. August 2005

PLISZKA 2009
PLISZKA, Julia: *Pervasive Gaming Framework*. 2009

Roussos 2006
ROUSSOS, George: *Ubiquitous and Pervasive Commerce: New Frontiers for Electronic Business*. Springer-Verlag, 2006

Salchow 2009
SALCHOW, Peter: *Pervasive Gaming Framework*. 2009

U. Hansmann u. Stober 2003
U. HANSMANN, M. S. N. I. Merk M. I. Merk ; STOBER, T.: *Pervasive Computing Handbook*. 2. Aufl. Springer-Verlag, 2003

vmware.com 2009
VMWARE.COM: *VMware ESX 3.5*. `http://www.vmware.com/files/de/pdf/esx_datasheet_de.pdf`. Version: Februar 2009