



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht

Oliver Köckritz

Kollaborationsspiel für
Collaborative Workspaces

Oliver Köckritz

Kollaborationsspiel für
Collaborative Workspaces

Projektbericht, 3. Semester, Projekt
im Studiengang Master of Computer Science
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Professor : Prof. Dr. Kai von Luck

Abgegeben am 1. März 2007

Inhaltsverzeichnis

1	Einleitung	2
2	Arbeitsweise	2
3	Orientierungsphase.....	3
3.1	Eigenentwicklung eines Tabletop mit Touchtechnologie	3
3.2	Überprüfung der Inhalte	4
3.3	Skizzen einer Anwendung	5
3.4	Techniken zur Verteilung	6
3.4.1	Verteilte Objektkommunikation	6
3.4.2	Szenengraphen.....	7
3.4.3	Croquet	7
3.5	Bewertung der Orientierungsphase	8
4	Implementationsphase.....	9
4.1	Aufteilung von Verteilung und Physikengine	9
4.2	OpenSG Tutorial	10
4.3	Pong-Szenario	10
4.4	Implementation	10
5	Ausblick und Bewertung	11
6	Literaturverzeichnis	12

1 Einleitung

Ein Computer ohne Betriebssystem funktioniert nicht, genauso wenig wie ein Collaborative Workspace (CW) ohne eigenes Betriebssystem oder Software, die diesen als CW nutzbar macht. Es gibt bekannte Ansätze wie BEACH [Tandler], COAST [Schümmer] oder iROS [JohansonDiss], um einen CW mit kollaborativen Eigenschaften zu füllen. Ein Framework oder eine Betriebssystemerweiterung sind die grundsätzlichen Ansätze. Um mich an einen neuen Ansatz der Windowsystemerweiterung unter technischen sowie inhaltlichen Aspekten heranzuarbeiten, habe ich mir im Projekt das Ziel gesetzt, eine Beispielanwendung für einen CW zu schreiben und die beiden Aspekte dabei so gut wie möglich abzudecken. Dieser Projektbericht beschreibt mein Vorgehen, meine Thesen und die Implementation einer auf das Minimum reduzierten Variante.

Ich beschäftige mich bei diesem Projekt nicht explizit mit einer Arbeitsumgebung, sondern vielmehr mit einer Lernumgebung, aber wie in [Köckritz] und [Schwabe] können Arbeitsumgebungen auch als Lernumgebungen und andersrum gesehen werden. Denn wenn ich in einer Gruppe etwas produziere, lerne meistens ich etwas und wenn ich etwas lerne, produziere ich häufig etwas.

2 Arbeitsweise

Das Projekt besteht aus drei Teilen. Einer der Teile begleitet den Aufbau des Collaborative Workspace Labors an der HAW-Hamburg und läuft parallel zu den anderen beiden Teilen. Die anderen beiden Teile folgen einander, wobei die Erfahrungen des zweiten Teils in den Dritten mit einfließen.

Grund für diese Aufteilung ist, dass der Laboraufbau abhängig von Lieferanten und anderen Querfaktoren ist. Der Zeitrahmen des Laboraufbaus war von Anfang an unklar und musste deshalb als parallel laufender Aufgabenbereich betrachtet werden. Um einer Rollen- und Aufgabenverteilung im Projekt und innerhalb der Projektgruppe Raum zu geben, wurde der Implementierungsteil zeitlich nach hinten verlagert und am Anfang eine Orientierung im Bereich möglicher technischer Umsetzungsformen und Thematiken vorgenommen.

Konkret wurde in den sequentiellen Teilen erst von außen nach innen gearbeitet. Das heißt, dass zunächst erste Ideen aufgeschrieben und in weiteren Schritten detaillierter überprüft

wurden, wobei sich während der ersten Evaluationsphase auch neue Aspekte für den ersten wie auch für den zweiten Teil ergeben haben. Als der Implementierungsteil begann, waren eigentlich alle inhaltlichen Fragen soweit geklärt, dass sie während des Projektzeitraumes keine Veränderungen in Bezug auf die Implementierung erforderten. In der Implementierungsphase habe ich von innen nach außen gearbeitet, also erst kleine Kernstücke erarbeitet, um sie immer mehr für das eigentliche Ziel auszubauen.

3 Orientierungsphase

3.1 Eigenentwicklung eines Tabletop mit Touchtechnologie

Neben den typischen Merkmalen eines Collaborative Workspaces, wie z.B. Powerwall, Whiteboard, Laptops, PDA und Auf- oder Rückprojektionen mit Interaktionsmöglichkeit, ist ein Tabletop meistens nicht nur physikalisch sondern auch methodisch zentral im Raum angeordnet. Dies ist ein wesentlicher Ansatz der Kollaboration in CWs. Menschen können gleichberechtigt mit eigenen, aber für alle sichtbaren Territorien zusammen arbeiten und etwas gemeinsames dabei entwickeln.

Um diese Kriterien zu erfüllen, muss der Tabletop groß genug sein, von allen Seiten erreichbar und es muss möglich sein, mit mehreren Menschen gleichzeitig an ihm zu interagieren. Bei Gesprächen zur Beschaffung des Equipments für den Collaborative Workspaces an der HAW-Hamburg stellte sich heraus, dass es schwer möglich sein würde, ein Gerät zu beschaffen, das diesen Anforderungen gerecht wird. Verschiedene Lieferanten wurden gefragt und letztlich war die Beschaffung eines Smartboards und eines passenden LCD-Monitors beschlossen worden. Ein Smartboard funktioniert zwar gut, aber es lässt kein gleichzeitiges Handeln von mehreren Menschen zu. Eine Beschreibung der Erweiterung auf Basis des Smartboards und eine umfangreiche Erörterung dieses Forschungsbereiches findet sich bei [Zachhuber].

Für die Powerwall und die Projektionen ist eine Interaktionsmöglichkeit in Form von Touchfähigkeit notwendig. Da dieser Aspekt nicht ganz gesichert war, überprüfte ich, ob es möglich ist, eine Ersatztechnologie selbst herzustellen. Diese Touchtechnologie soll vielen Nutzern ermöglichen, gleichzeitig auf einem Display zu interagieren. Eine eventuelle Lokalisation der interagierenden Personen im Raum wäre damit auch möglich. Das Überprüfen der Möglichkeit, diese Technologie zu implementieren, war erfolgreich. Ein experimenteller Probeaufbau mit Signalmessung endete sehr vielversprechend. Verschiedene Positionierungen lieferten ein signifikant verschiedenes klares Signal. Unklar ist noch, ob entsprechende Materialien einfach zu beschaffen sind oder ob sie extra

produziert werden müssen. Im letzteren Fall müsste eine genauere Überprüfung stattfinden, da sonst der Aufwand der Produktion im Falle eines Scheiterns nicht gerechtfertigt wäre.

Fazit der Überprüfung von der erdachten Technik ist: Eine Eigenentwicklung einer Touchtechnologie durch Positionserkennung des Fingers auf dem Bildschirm mit den entsprechenden Materialien ist möglich. Nicht oder nur schwer realisierbar ist die Anwendung dieser Technologie, wenn Benutzer mit mehreren Fingern arbeiten wollen. Gleichzeitiges Bereitstellen der Funktionalitäten von Drag und Drop sowie Button Press ist durch Implementierung eines entsprechenden Treibers ebenfalls möglich. Die Möglichkeit von Left- und Right-Klick scheint mir in diesem Fall eine nicht so einfach differenzierbare Usereingabe zu sein.

Als Alternative zur Beschaffung des Tabletop habe ich zeichnerisch einen Tisch entwickelt, der mit zwei Beamern, einer Rückprojektionsplatte und einer Umlenkung durch Spiegel funktioniert.

3.2 Überprüfung der Inhalte

Um ein Szenario in einem CW zu schaffen, das gegebenenfalls leicht zu implementieren, zu nutzen und zu präsentieren ist, habe ich mir Gedanken über ein kollaboratives Spiel zur Schulung von so genannten Sekundärtugenden gemacht. Zur Aneignung gegebener Methoden und Erkenntnissen habe ich mir die Inhalte der Bücher „Metaphern – Schnellstraße, Saumpfade und Sackgassen des Lernens“ [Schödlbauer] und „Kooperative Abenteuerspiele 2“ [Gilsdorf] erarbeitet. Starken Bezug habe ich auf das Kapitel „Metaphorisches Lernen und moderne Theorien der Selbstorganisation“ aus dem Buch „Metaphern“ gelegt. Das Buch „Kooperative Abenteuerspiele 2“ habe ich lediglich zur Ideenfindung und als Lieferant abstrakter Beispiele für die Anwendung im CW genutzt.

Zur Reflexion habe ich mich mit drei Pädagogen aus verschiedenen Bereichen zu einem Gespräch getroffen, um die Grundidee zu erörtern. Das erste Gespräch führte ich mit einem Pädagogen mit Lehrhintergrund und mittlerweile leitender Tätigkeit im Bereich Neue Medien in der Schule. Mit Zustimmung und Verständnis wurde die Idee des selbstorganisierten Lernens von Sekundärtugenden in Bezug auf Gruppenarbeit und –dynamik mit Hilfe eines CW aufgenommen. Da es noch keine konkrete Anwendung vermittelbar war, konnte hier leider nicht auf Details eingegangen werden.

Das zweite Gespräch führte ich mit einer Lehrerin der Grundschule. Hier ergaben sich konkrete Beispiele, da sich herausstellte, dass sie den Kindern Gruppenspiele anbietet, die selbstorganisierten Lernen von Sekundärtugenden in Bezug auf Gruppenarbeit fördern. Hier stellte sich heraus, dass es nicht unbedingt nötig ist Computertechnologie einzusetzen, um dieses pädagogische Ziel zu erreichen. Betrachtet man aber, dass es sich in diesem Fall um sehr viel Engagement seitens der Lehrerin handelt, indem sie eine nicht standardisierte Lehrmethode anwendet, könnte hier eine computergestützte Substitution durchaus sinnvoll sein.

Das dritte und letzte Gespräch fand mit einer Erlebnispädagogin statt, die die Grundidee kollaborativen Arbeitens an sich positiv bewertet, jedoch dem Einsatz von Computertechnologie in diesem Fall kritisch gegenüber steht. Ihrer Meinung nach findet ein sehr starker Anteil der Arbeit in diesem Gebiet auf einer zwischenmenschlichen Ebene auch zwischen Pädagoge und Kind statt. Eine Anpassung an verschiedenste Situation sollte einfach umsetzbar sein und kann hier nicht von einem statischen nicht intelligenten Computer ohne Einfühlungsvermögen geleistet werden.

Diese gezielt ausgesuchten Gespräche sind wissenschaftlich nicht repräsentativ, wohl aber geben sie einen Hinweis darauf, dass es in der Pädagogik einen Trend zu Partizipation und kollektiver Intelligenz gibt. Es bleibt fraglich, ob bei einer fest vorgegebenen Struktur das Ziel der Partizipation noch einen Platz findet. Gehen wir aber davon aus, dass der Computer seinen Platz in mitten der Gesellschaft gefunden hat und somit auch die damit einhergehenden Verhaltensmuster, wäre eine Auflösung der alten und gebräuchlichen Interaktionsmuster in Bezug auf Computertechnologie sinnvoll, da es die bindende Klammer auflösen könnte und neue Räume eröffnet.

3.3 Skizzen einer Anwendung

Während des Projekts ist auch eine Skizze eines verteilten Tetris entstanden. Jedes Kind hat seinen eigenen kleinen Bereich in dem es eine Teilaufgabe individuell lösen muss. Diese Lösung kann durch die Anordnung auf dem Tabletop gut von anderen eingesehen werden, so dass Hilfestellungen durch andere Kinder möglich sind. Alle Kinder am Tisch arbeiten mit ihren individuellen Lösungen Teilstücke für die gemeinsame Lösung aus. Die Art der Teilstücke für die gesamte Lösung muss ohne Computerunterstützung verhandelt werden. Ebenfalls ohne Technik muss die Ressourcenverteilung für den individuellen Arbeitsplatz verhandelt werden. Alle zu verteilenden Puzzleteile sind im Ressourcentopf personenneutral und müssen bei Bedarf den jeweiligem Kindern nähergebracht werden, da es dieses eventuell nicht erreichen kann. Nach der individuellen Zuordnung werden sie farblich markiert. Diese Markierung bleibt bis zum Schluss. Eine Gleichverteilung der Markierungen im Gesamtergebnis führt zu einer höheren Bewertung. Die Kinder müssen also Ressourcen teilen, Lösungen miteinander abgleichen und gegenseitig darauf achten, dass alle ein ähnliches Arbeitspensum haben. Um einem Ausschluss von Kindern vorzubeugen, wird ein Ranking eingeführt. Hat eine Gruppe ein gut verteiltes Gesamtranking, so führt auch dies wieder zu einer höheren Bewertung. Interesse der Gruppe muss nun sein, dass alle aus der Gruppe einen gleichen Anteil zum Ergebnis liefern und dass die Gruppe aus allen Bereichen des Rankings besteht.

3.4 Techniken zur Verteilung

Bei der Verteilung der Objekte und Daten in einem CW habe ich mir verschiedene Techniken angeschaut um zu entscheiden, auf welcher Basis ein Prototyp für eine kollaborative Anwendung fußen kann.

Bei einer verteilten Kommunikation über Sockets gibt es zwar keine besonderen Voraussetzungen, es ist schnell und generell verfügbar, aber es muss ein eigenes Protokoll zur Kommunikation definiert werden und damit auch alle Elemente wie Protokollprüfung, Parameter- und Ergebnisübergabe sowie Fehlerbehandlung, deshalb habe ich diese Variante gleich ausgeschlossen.

3.4.1 Verteilte Objektkommunikation

Remote Method Invocation (RMI)

RMI unterstützt eine einfache Kommunikation zwischen Java-Objekten, die in unterschiedlichen Java-Laufzeitumgebungen existieren. Für den Entwickler gibt es fast keinen Unterschied zwischen lokalen und verteilten Objekten. Deshalb kann mit RMI bei der Entwicklung von verteilten Anwendungen fast genauso gearbeitet werden, wie bei der herkömmlichen Entwicklung von lokalen Anwendungen.

Common Object Request Broker Architecture (CORBA)

CORBA's verteilte Kommunikation basiert wie RMI auf der Kommunikation von verteilten Objekten. Zur Kommunikation wird das IIOP-Protokoll benutzt, wodurch jede auf CORBA basierende Anwendung unabhängig von Implementierungssprache, Betriebssystem, Rechner und Netzwerk mit anderen CORBA Anwendungen kommunizieren kann. Damit verteilte Objekte transparent miteinander kommunizieren können, benötigt CORBA einen Object Request Broker (ORB). Dieser erlaubt auch das Auffinden anderer Objekte innerhalb des verteilten Systems. Deshalb muss auf jedem Rechner innerhalb des verteilten Systems ein ORB laufen.

Vergleich: CORBA vs RMI

Vergleichen wir RMI und CORBA, so stellen wir fest, dass beide plattformunabhängig sind. RMI ist jedoch sprachgebunden an Java, während CORBA sprachunabhängig ist. CORBA besitzt keine Garbage Collection, weswegen es im Gegensatz zu RMI selbst implementiert werden muss. RMI kann dynamisch neue Klassen integrieren und leicht implementiert werden, da es keine zusätzliche Beschreibungssprache benötigt. Zusätzlich spricht der benötigte Object Request Broker (ORB) auf jedem beteiligten Rechner gegen CORBA. Wenn zwischen CORBA und RMI entschieden werden müsste, würde die Wahl wegen der Schwergewichtigkeit von CORBA auf RMI fallen. Der Vorteil der Sprachunabhängigkeit von CORBA würde hier nicht zum Tragen kommen, da jegliche Funktionalität ohnehin

neuentwickelt werden müsste und dies auch in Java geschehen kann. Diese Entscheidung wird dadurch bestätigt, dass das Vorbild iROS auch mit Java/RMI verwirklicht wurde.

3.4.2 Szenengraphen

Mit einem Szenengraphen [SG] lassen sich 3D-Szenen verteilt darstellen. Dabei werden nur die sichtbaren Teile gerendert. Die Struktur des Graphen hält implizit die Informationen über die Abhängigkeiten der zu verteilenden oder sich verändernden Graphenteilstücke. Da nur die Veränderungen des Graphen synchronisiert werden müssen, kann der Szenengraph effektiv über ein Netzwerk verteilt werden. Um nur einen bestimmten Teil des Gesamtgraphen sichtbar zu machen, können die Projektionspositionen verändert werden.

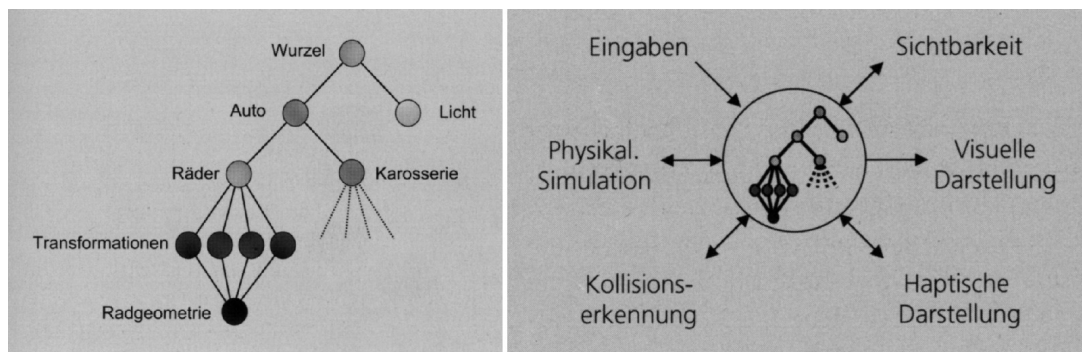


Abbildung 1: (links) Beispiel eines Szenengraphen; (rechts) Möglichkeiten zur Nutzung und Erweiterung eines Szenengraphen. [Bilder aus Reiners]

Wie in Abbildung 1 illustriert kann der Szenengraph zur Manipulation durch eine Eingabe, zur Verteilung und zur visuellen Darstellung genutzt werden. Auch andere Darstellungsformen, wie z.B. eine haptische Darstellung, können möglich gemacht werden. Zur Berechnung des Bewegungsverhaltens können außerdem spezielle Bewegungsverhalten wie eine physikalische Simulation genutzt werden. Nicht lineare Regeln zur Manipulation wie z.B. Kollisionserkennung sind ebenfalls möglich.

3.4.3 Croquet

Croquet ist eine Kollaborationsumgebung, in der Clients dynamisch ein- und austreten können. Jeder Teilnehmer besitzt normalerweise sein eigenes Ein- und Ausgabemedium und wird in der in 3D dargestellten Welt durch einen Avatar repräsentiert. Alles in dieser kollaborativen Welt wird durch Objekte dargestellt und jeder kann diese Objekte manipulieren. Croquet setzt auf die virtuelle Maschine (cross-plattform) Squeak auf und besteht neben der dreidimensionalen Darstellung, die mit Szenengraphen realisiert ist, aus einem Peer to Peer Netzwerk von Clients und Server.

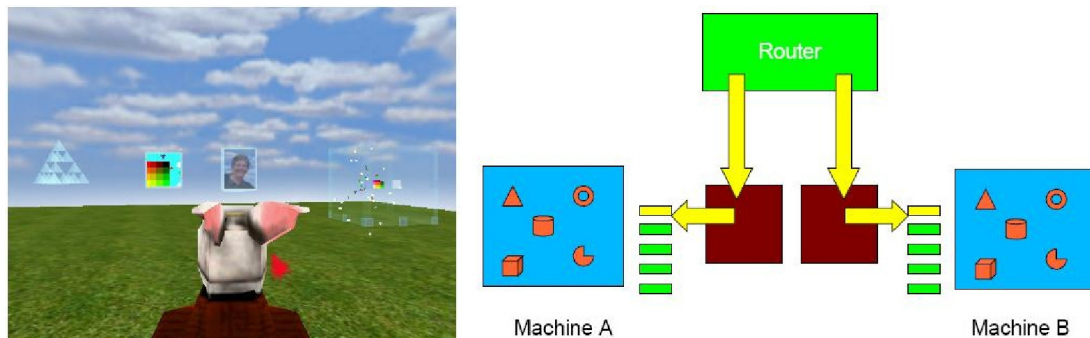


Abbildung 2: (links) Blick des Avatars auf die Welt (Insel). Schwebende Objekte können auch Portale oder Anwendungen sein; (rechts) Der Router synchronisiert Ereignisse und sendet sie an die Message-Queues der angemeldeten Repliken der Insel. [Bilder aus Smith]

Croquet beinhaltet eine Architektur, die eine eindeutige Zeit innerhalb eines laufenden Systems sicherstellt. Jede Veränderung wird mit Hilfe dieser Zeit rekonstruierbar und eindeutig ausführbar. Jede Veränderung der Welt ist deterministisch und wird in dem Szenengraphen und den dazu gehörigen Objekten gespeichert. Anhand dieser Welt wird dann die Darstellung gerendert. Vergleichen wir die Bestandteile des verteilten 3D-Desktops mit Croquet, so ist der Mauszeiger der Avatar, die Anwendungen (Windows) die Objekte und die Welt (Insel) der verteilte 3D-Desktop.

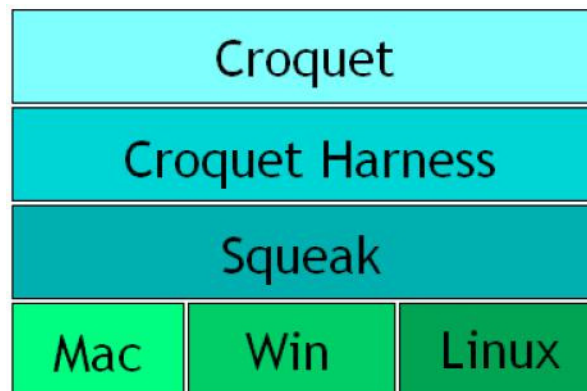


Abbildung 3: Struktureller Aufbau von Croquet

3.5 Bewertung der Orientierungsphase

RMI war die erste Wahl bei der Entscheidung für die anzuwendende Kommunikationsstruktur zur Implementierung einer ersten kleinen verteilten Anwendung. Um Protokolle und Definitionen musste sich nicht gekümmert werden und die Anwendung wäre leichtgewichtig und damit auf Handhelds lauffähig.

Da aber innerhalb des Projektes auch andere eine Verteilung anstrebten und es sinnvoll war, eine allgemeine Verteilung auf Basis von graphischen Objekten anzustreben, verlagerte sich die Diskussion um eine Abstraktionsebene höher hin zu Szenengraphen. Java3D und OpenSG waren die beiden Kontrahenten. Da ich mich mit beiden Systemen erneut vertraut machen musste, Philipp Rossberger aber in seinem Projekt einer physikbasierten Interaktion schon eine geeignete Physik-Engine (ODE) gefunden hatte, entschieden wir uns für OpenSG. Ein weiterer Grund hierfür ist, dass im Bereich der Szenengraphen die meisten Systeme in C++ geschrieben sind und Java3D mit Java sowie Croquet mit Smalltalk im Feld der Szenengraphen eher eine Ausnahme darstellen.

Der Tabletop mit Touch sowie die Skizze einer Anwendung dienten hier nur zur Ideenfindung bzw. als Referenz zur Findung der ersten Anwendung und der dazu genutzten Technik. Die Ergebnisse aus diesen drei Bereichen können aber später wieder verwendet bzw. aufgenommen werden. Als Beispielanwendung für ein nahes Ziel habe ich mich zu diesem Zeitpunkt für ein verteiltes Pong-Spiel entschieden.

4 Implementationsphase

Die Implementierungsphase besteht aus den drei parallel laufenden Teilen Arbeitsaufteilung, Tutorial und Szenario gefolgt von einem finalen Teil der Implementierung der Anwendung und damit der verteilten Beispielanwendung als Referenztechnologie.

4.1 Aufteilung von Verteilung und Physikengine

Ein Teil des Gesamtprojekts ist die Evaluierung einer Physikengine zur Benutzung physikbasierter flüssiger Interaktion von Philipp Rossberger. Da wir uns gemeinsam entschieden hatten, OpenSG als Grundlage unseres Wirkens zu nehmen, wurde der Bereich Einbindung und Nutzung der Physikengine Philipp zugeschrieben. Ich habe den Bereich der Verteilung einer Anwendung mit OpenSG auf verschiedenen Rechnern übernommen. Mausinteraktion und Grafik konnte und sollte nicht getrennt werden, so dass dieser Teil von beiden bearbeitet werden musste. In der Auseinandersetzung mit der Entwicklungsumgebung Visual C++ mit allem was dazu gehört konnten wir auch auf Synergieeffekte mit Christian Fischer hoffen, da dieser seinen Projektteil der multimodalen Interaktion mit einem „Grabstick“ auch auf Visual C++ fußen lassen wollte.

4.2 OpenSG Tutorial

Erste Aufgabe war es, das Tutorial „OpenSG Starter Guide“ [OpenSGTut] nachzuvollziehen und Beispiele aus diesem zu implementieren. Das Tutorial ist zwar für Linux, Windows und MacOS ausgelegt, wohl aber sind die Beispiele größten Teils unter Linux ausprobiert worden, so dass nicht alles auf Anhieb funktionierte. Die Teile FirstApp, Basics, NodeCores, Windows und Clustering wurden erfolgreich nachvollzogen. Clustering ist hier wohl das Interessanteste, da es in diesem Kapitel um die Verteilung geht, allerdings nur sehr oberflächlich. Ein wirkliches Verständnis erlangt man dadurch nicht. Nur eine erste Anwendung auf Basis des Beispiels kann hiermit implementiert werden. Die anderen Teile sind nur Mittel zum Zweck, um ein Gefühl für OpenSG zu bekommen. Es stellte sich auch heraus, dass OpenGL, auf das OpenSG aufbaut, auch mitberücksichtigt werden muss, da auch ein Verständnis von OpenGL erforderlich ist.

4.3 Pong-Szenario

Es gibt bereits verschiedene Formen von Pong, und Pong wird immer wieder dazu genutzt, einfache Thesen zu testen oder zu zeigen. Die Web-Seite des Computerspiele-Museums [Pongmythos] des Fördervereins für Jugend und Sozialarbeit e.V. aus Berlin zeigt dies deutlich. NetPong der Uni-Hannover, Susipong [Susipong] aus Karlsruhe und 3DPong [3DPong] sind hier nur einige gute Beispiele. Mein Szenario lehnt sich vor allem an das 3DPong Szenario an. Es soll möglich sein, aus der Perspektive des eigenen Schlägers über das Spielfeld zu schauen. Der Gegenspieler befindet sich am anderen Ende an einem anderen Rechner im Netz. Ist dieses Szenario mit OpenSG implementiert, können weitere und sinnvollere Szenarien für einen CW einfach implementiert werden.

4.4 Implementation

Zur Implementation wurde der geschriebene Code aus dem Tutorial wiederverwendet. Leider lieferte das Tutorial keine ausreichende Information um den Code selber zu erstellen und wirklich zu verstehen, so dass sehr viel auf Foren und Mailinglisten wie [listsOpenSG] nach den richtigen Antworten gesucht werden musste.

Zu beachten ist, dass in den meisten Implementationen mit OpenSG der Idle-Event genutzt wird, um einen ruckelfreien Programmfluss beim Rendering zu bekommen. Dies ist ein fataler Fehler, denn jegliche nicht genutzte Prozessorleistung wird hiermit zum Rendering genutzt. Dies hat zur Folge, dass bei hoher Prozessorleistung jede Szene so häufig wie möglich gerendert wird. Es hat keinen Sinn, jede Szene über 100 mal pro Sekunde neu zu berechnen und gegebenenfalls auch noch Synchronisationsmitteilungen übers Netz zu

verschicken. Es ist Ratsam, eine Implementierung von einer festen Frame/Sec Rate wie sie bei Computerspielen üblich ist vorzunehmen.

Die Implementierung von Multiusereingaben ist mir bis zum heutigen Zeitpunkt leider noch nicht gelungen. Da bei dem genutzten Konstrukt davon ausgegangen wird, dass Eingabe und Ausgabe auf verschiedenen Rechnern bzw. Windows erfolgen.

Der jetzige Programmcode ist noch unstrukturiert und nicht dokumentiert. Es bedarf also noch einiger Nacharbeiten, um Teile des Codes gut lesbar und wiederverwendbar zu machen.

5 Ausblick und Bewertung

Leider kann auf keiner Ebene der Vorgehensweise ein wirklicher Erfolg gezeigt werden. Nur Teilerfolge sind zu verzeichnen. So hat sich die mögliche Funktionsfähigkeit des Tabletops mit Touchtechnologie bewiesen, aber die Schwierigkeiten zur Beschaffung der Materialien hat sich bestätigt. Auch gab es Bestätigungen für die Umsetzung einer kollaborativen Lernsoftware. Aber auch hier zeigte sich, dass die Umsetzung weit komplexer ist als die reine Ideenfindung.

Die Nutzung von OpenGL hat gezeigt, dass es eine komplexe Klassenbibliothek ist, welche besser dokumentiert sein könnte, was zur Folge hat, dass auch am Ende des Projekts fragen offen blieben wie z.B. ob sich OpenGL wirklich dazu eignet, einen verteilten 3D-Desktop mit OpenGL zu implementieren. Die Annahme und das Wissen, dass es sich mit anderen Systemen genauso verhält und auch dort die gleichen Erweiterungen getätigt werden müssten, führt zu einem ähnlichen Ergebnis wie bei der Überprüfung: Es scheint vielversprechend, es muss aber noch viel getan werden bzw. es gibt keine Garantien für das Gelingen.

Das verteilte Pong-Spiel werde ich noch nach Beendigung des Projektes fertig stellen, damit die investierte Arbeit noch sichtbar und bewertbar wird. Der Weg innerhalb der Masterthesis zum verteilten 3D-Desktop könnte auch direkter zum Ziel führen, aber die im Projekt eingeschlagene Arbeitsweise hat sich für mich am Ende klar bestätigt. Es gäbe die Möglichkeit, die Orientierungsphase zu verkürzen bzw. zu streichen, aber ich vermute, dass dieses zu anderen Verzögerungen geführt hätte.

6 Literaturverzeichnis

- [3DPong] Janecek, A. and Hlavacs, H. 2005. Programming interactive real-time games over WLAN for pocket PCs with J2ME and .NET CF. In Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support For Games (Hawthorne, NY, October 10 - 11, 2005). NetGames '05. ACM Press, New York, NY, 1-8. DOI= <http://doi.acm.org/10.1145/1103599.1103603>
- [Gilsdorf] Rüdiger Gilsdorf, Günter Kistner: Kooperative Abenteuerspiele 2. Kallmeyer, Seelze-Velber 2001
- [JohansonDiss] Bradley Earl Johanson; Application Coordination Infrastructure for Ubiquitous Computing Rooms, Dissertation, Department of Electrical Engineering and the Committee on Graduate Studies of Stanford University, December 2002
- [Köckritz] Oliver Köckritz; Geschichte und Konzept von Collaborative Workspaces, Seminararbeit an der HAW-Hamburg, 2006
- [listsOpenSG] <http://www.mail-archive.com/opensg-users@lists.sourceforge.net/> (Zugriff 24.2.2007)
- [OpenSGTuto] <http://www.opensg.org/doc-1.6.0/TutorialIntroduction.html> (Zugriff 20.10.2006)
- [Schödlbauer] Cornelia Schödlbauer, F. Hartmut Paffrath, Werner Michl; Metaphern – Schnellstraßen, Saumpfade und Sackgassen des Lernens / Internationaler Kongreß „erleben und lernen“; Praktische Erlebnispädagogik, Ziel Verlag, Augsburg 1999
- [Schümmer] Jan Schümmer, Till Schümmer, Christian Schuckmann, COAST – Ein Anwendungsframework für synchrone Groupware, Conference Proceedings for the „Net.ObjectDays 2000“, <http://www.opencoast.org> (Zugriff 20.2.2007)
- [Schwabe] G. Schwabe, C. Filk, and M. Valerius. Warum Kooperation neu erfinden? -- zum Beitrag der CSCW-Forschung fürs kollaborative e-learning. In H. U. Buhl, A. Huther, and B. Reitwiesner, editors, Information Age Economy, pages 381-394. Physika, Heidelberg, 2001.
- [Susipong] <http://www.susipong.com/> (Zugriff 23.2.2007)
- [Tandler] Peter Tandler, The BEACH Application Model and Software Framework for Synchronous Collaboration in Ubiquitous Computing Environments, Submitted to JSS, special issue on UbiTools v4.2, 26.02.2003
- [Zachhuber] Doris Zachhuber; HoloTouch: Handgestenbasierte Interaktion mit einem auf optischem Stereo-Tracking basierenden Bildbrowser unter Verwendung eines transparenten Wanddisplays; DIPLOMARBEIT, eingereicht am Fachhochschul-Masterstudiengang Digitale Medien in Hagenberg im Juni 2006; <http://www.zachhuber.net/HoloTouchDorisZachhuber.pdf> (Zugriff: 28.2.2007)