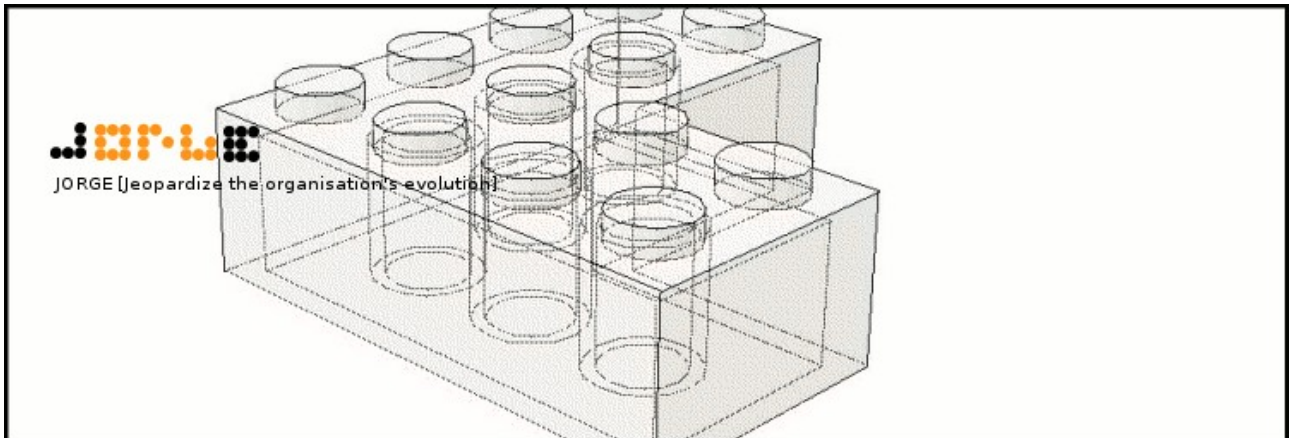




Berner Fachhochschule

Hochschule für Technik und Informatik HTI



# Schlussbericht

## [Jorge - Lego MindStorms]

Autoren	Nik Lutz [I3STW, lutzn@hti.bfh.ch] Stefan Feissli [I3STW, feiss@hti.bfh.ch] Christof Seiler [I3STW, seilc@hti.bfh.ch]
Version	1.0
Datum	17.06.05
Status	Freigegeben
webseite	jorge.hta-bi.bfh.ch
Projektauftraggeber	Claude Fuhrer
PM-Coaching	Frank Helbling
Projektarbeit	Sommersemester 2005, HTI Biel, Abt. Informatik



# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>3</b>
1.1 Zweck des Dokuments	3
1.2 Erwähnte Dokumente und Referenzen	3
<b>2 Soll-Ist-Vergleich</b>	<b>3</b>
2.1 Resultate	3
2.2 Termine	4
2.2.1 Projekt Dokumentation	5
2.2.2 Produkt Dokumentation	5
2.2.3 Produkt	5
2.3 Aufwände	6
2.4 Kosten	6
2.4.1 Begründungen zur Kostensituation des Projektes	7
<b>3 Erkannte Probleme</b>	<b>7</b>
3.1 Gelöste Probleme	7
3.2 Ungelöste Probleme	9
<b>4 Zielerreichung</b>	<b>9</b>
<b>5 Gewonnene Erkenntnisse</b>	<b>10</b>
5.1 Generell	10
5.2 Zum Modell Hermes	10
5.3 Technologie	11
5.3.1 Ogre	11
5.3.2 Ode	11
5.3.3 Blender	11
5.3.4 Emulator	11
5.4 Infrastruktur	11
5.5 Organisation	12
5.6 Projektarbeit/Teamarbeit	12
5.7 Betreuung Fachdozent + PM Coach	12
<b>6 Folgerungen für die Diplomarbeit/künftige Projektarbeit</b>	<b>13</b>
<b>7 Persönliche Meinungen</b>	<b>13</b>
7.1 Christof Seiler	13
7.2 Niklaus Lutz	13
7.3 Stefan Feissli	14
<b>8 Schlussfolgerungen</b>	<b>14</b>
8.1 was lief gut	14
8.2 was sollte beim nächsten Mal besser gemacht werden	14

# 1 Einleitung

## 1.1 Zweck des Dokuments

Mit diesem Dokument schliessen wir die die Arbeiten im Rahmen der Semesterarbeit für JORGE ab. Ziel dieses Dokuments ist eine Zusammenfassung der Erfahrungen und der erreichten Ziele die wir während unserer Projektarbeit gemacht haben.

## 1.2 Erwähnte Dokumente und Referenzen

Nummer	Name	Beschreibung	Version	Ort
1	Projektplan	Projektplan der Semesterarbeit	1.0	<a href="http://jorge.hta-bi.bfh.ch">http://jorge.hta-bi.bfh.ch</a>
2	Ogre	3D Framework	1.1	<a href="http://www.ogre3d.org">http://www.ogre3d.org</a>
3	Ode	Collision Detection Library	0.5	<a href="http://www.ode.org">http://www.ode.org</a>
4	OgreOde	Ogre wrapper for Ode	1.0.0	<a href="http://www.green-eyed-monster.com">http://www.green-eyed-monster.com</a>
5	LeJos	Firmware für Legoroboter	2.1.0	<a href="http://lejos.sourceforge.net/">http://lejos.sourceforge.net/</a>
6	Cygwin	*nix Umgebung für windows	1.5.17-1	<a href="http://www.cygwin.com">http://www.cygwin.com</a>
7	Boost	Threading Library	1.32.0	<a href="http://www.boost.org">http://www.boost.org</a>
8	WxWindows	GUI Framework		<a href="http://www.wxwindows.org">http://www.wxwindows.org</a>
9	Eclipse	IDE		<a href="http://www.eclipse.org">http://www.eclipse.org</a>
10	Code Forge	IDE		<a href="http://www.codeforge.com">http://www.codeforge.com</a>
11	MS Visual Studio	IDE		<a href="http://www.microsoft.com">http://www.microsoft.com</a>

# 2 Soll-Ist-Vergleich

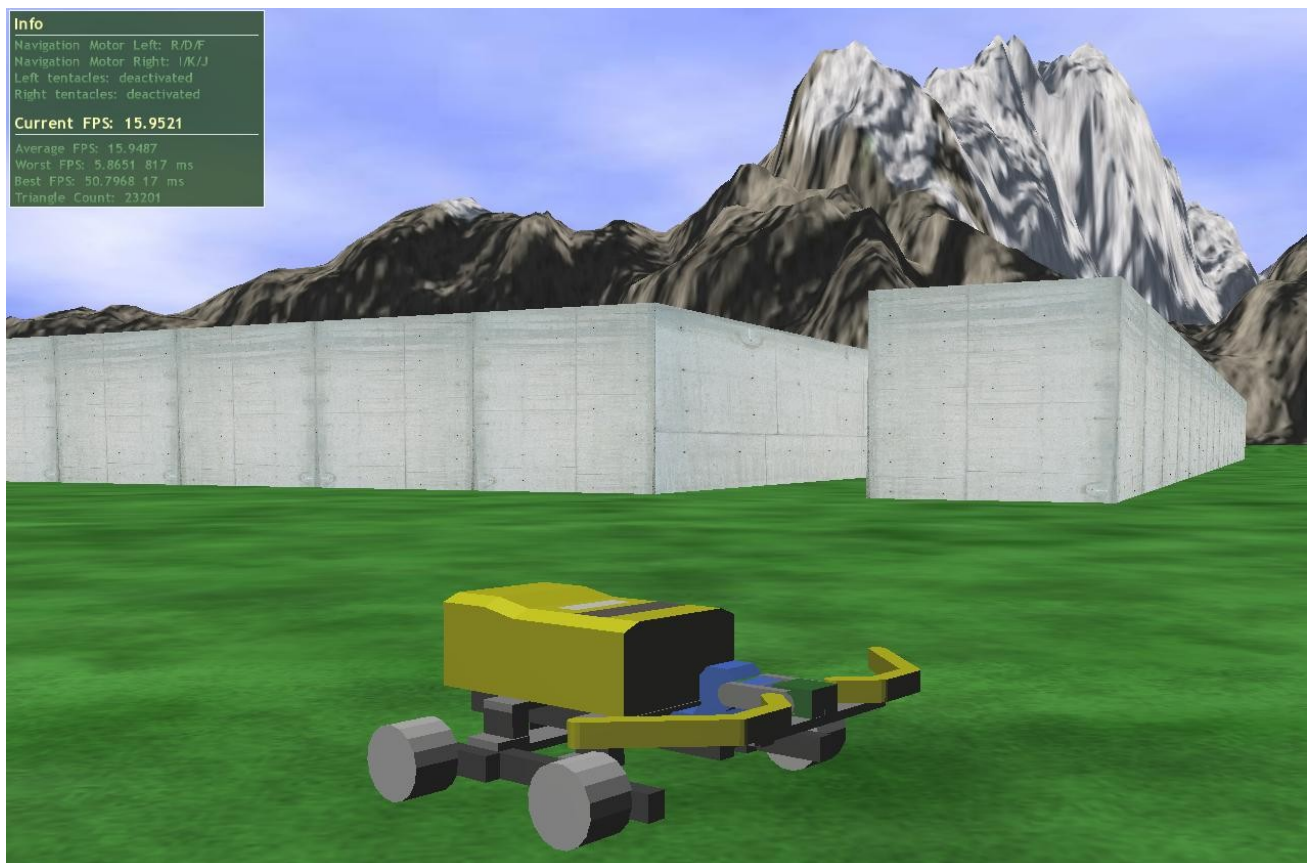
## 2.1 Resultate

wir waren uns bewusst, eine sehr anspruchsvolle und risikobehaftete Aufgabe gestellt bekommen zu haben. Durch den Einsatz von C++ anstelle von Java wagten wir uns in eine für uns eher unbekannt Umgebung vor, insbesondere für grosse Projekte. Zudem hatten wir den Anspruch JORGE für den Einsatz unter verschiedenen Betriebssystemen zu entwickeln, was weitere Probleme mit sich brachte.

Die aktuelle Version von JORGE läuft unter Linux und windows. An der Kommandozeile kann dem Programm als Argument ein Dateiname übergeben werden. Wenn die angegebene Datei existiert und ein gültiges LeJos Kompilat enthält, können über einen Dialog bevorzugte Anzeige-Optionen gewählt werden. Danach startet das Hauptfenster von JORGE.

Das Fenster zeigt eine weitläufige Landschaft mit Hügeln und Bergen. Von oben fällt dann ein Roboter ins Bild, der nach belieben durch die Landschaft gesteuert werden kann. Mit einem bestimmten Tastendruck kann das zuvor geladene Program gestartet werden. Somit wird die Kontrolle des Roboters dem geladenen Programm übergeben.

Gegenwärtig können LeJos Programmen die beiden Motoren des virtuellen Roboters für die Steuerung nutzen. Die Motoren treiben vier Räder an, wobei die Steuerung wie bei einem Panzer funktioniert. Prallt der Roboter in der virtuellen Landschaft mit seinen Fühlern gegen eine Mauer des Labyrinth's, werden diese Information dem Emulator übergeben. Falls das gewählte Program auf die Sensor-Ereignisse mit Ein- oder Auschalten eines Motors reagiert, bewegt sich der Roboter entsprechend. All diese Ereignisse werden auf dem Bildschirm angezeigt.



## 2.2 Termine

Die nachstehenden Tabellen zeigen die Ergebnisse und deren Abschluss gemäss Ergebnisstrukturplan des Projektplanes [2]. Alle Ergebnisse konnten termingerecht erstellt und abgegeben werden.

### 2.2.1 Projekt Dokumentation

PM Dokumente	Gepplanter Abschluss	Abgegeben
Projektantrag	11.03.2005	11.03.2005
Projekthandbuch	15.04.2005	15.04.2005
Projektplan	15.04.2005	15.04.2005
Abschlussbericht	17.06.2005	17.06.2005
<b>Auftraggeber Dokumente</b>		
Pflichtenheft	15.04.2005	15.04.2005
<b>Voranalyse Dokumente</b>		
Lösungsvorschläge	17.06.2005	17.06.2005
Systemziele	17.06.2005	17.06.2005
<b>Konzept Dokumente</b>		
Systemdesign	17.06.2005	17.06.2005
Systemarchitektur	17.06.2005	17.06.2005

### 2.2.2 Produkt Dokumentation

Benutzer	Gepplanter Abschluss	Abgegeben
Anwendungsbuch	17.06.2005	17.06.2005
<b>Engineering</b>		
API Dokumentation	17.06.2005	17.06.2005
Betriebshandbuch	17.06.2005	17.06.2005

### 2.2.3 Produkt

Applikation	Gepplanter Abschluss	Abgegeben
Prototyp Jorge	17.06.2005	17.06.2005

## 2.3 Aufwände

Der Aufwand wurde von uns relativ gut eingeschätzt. Anstatt der kalkulierten 900 Stunden mussten wir 960 Stunden aufwenden. Dadurch steigen die Personalkosten bei einem Stundenansatz von 150 CHF, für einen Studenten, von CHF 147'000 auf 156'000. Die Schlusskostenabrechnung kann dem Punkt 2.4 entnommen werden.

Phase	Bemerkungen	Geschätzter Aufwand [Stunden]	Effektiver Aufwand [Stunden]
Initialisierung	Projekt analysieren und Projektvisionen entwickeln	130	135
Voranalyse	Systemziele werden ausformuliert und festgelegt, die Systemkomponenten werden evaluiert und eine Lösungsaswahl wird erstellt	140	180
Konzept	Konzept und Systemdesign werden entwickelt	160	165
Realisierung	Prototyp wird entwickelt	410	410
Abschluss	Projekt abschliessen und alle Dokumentationen fertigstellen	60	70
<b>Total</b>		<b>900</b>	<b>960</b>

## 2.4 Kosten

Teilkosten	Budget [CHF]	Effektiv [CHF]
Personalkosten	147'000	156'000
Sachkosten	19'900	19'900
<b>Nettokosten</b>	<b>166'900</b>	<b>175'900</b>
Risikozuschlag 10%	16'690	0
<b>Total Projektkosten</b>	<b>183'590</b>	<b>175'900</b>
Gewinn 0%	0	0
<b>Bruttokosten</b>	<b>183'590</b>	<b>175'900</b>

### 2.4.1 Begründungen zur Kostensituation des Projektes

Obwohl wir mehr Arbeitsstunden als budgetiert aufgewendet haben, konnte dieser Mehraufwand durch den Risikozuschlag gedeckt werden. So liegen wir im Rahmen der budgetierten Kosten.

Diese Abweichung ist durch diverse Probleme im Bereich der benutzen Softwaretools entstanden. Genaue Angaben zu diesen Problemen finden sie im nächsten Kapitel "Erkannte Probleme".

## 3 Erkannte Probleme

### 3.1 Gelöste Probleme

Problem	Geetroffene Massnahmen	Status
Alle OpenSource IDE's für Linux entsprachen nicht unseren Erwartungen.	Erwerb von Lizenzen für CodeForge. Ermöglichte effizientes Arbeiten.	Gelöst
Fehler in CodeForge (Abstürze)	Kontakt mit Entwicklern von CodeForge. Einsatz von Debug Libraries, um bei der Fehlersuche zu helfen.	CodeForge hat die Probleme jeweils gelöst, und uns die aktualisierten Versionen zugänglich gemacht
Geeignete Makefiles für Linux erstellen mit den etablierten werkzeuge autoconf und automake führte nicht zu einem befriedigenden Ergebnis.	Erstellen der Makefiles mit CodeForge und anpassen von Hand (absolute Pfade durch relative ersetzen)	Diese Lösung funktioniert, fehlt aber eine Bibliothek auf dem System des Benutzers erhält der Benutzer unklare Fehlermeldungen.
Das mit Hilfe des CAD Tools (Leocad) erstellte 3d-Lego Modell war viel zu gross (~65 MB).	Modell mit Hilfe von Blender (Object Modeling Tool) vereinfachen. Finale Grösse ~1MB.	Gelöst, das so erstellte Modell wird in der Applikation eingesetzt und entspricht unserer Erwartung in Bezug auf Aussehen und Funktionalität.
Blender nicht gut geeignet um 3D Landschaften zu generieren	Erwerb einer Lizenz für Freeworld3D um die 3D Landscht zu modellieren. Danach aufwändige Konvertierung in ein geeignetes Format, um mit Terragen verschiedene Texturen auf die oberflächen zu legen.	Gelöst. Der dafür erarbeitete Ablauf eignet sich gut um Landschaften zu generieren.

<b>Problem</b>	<b>Geetroffene Massnahmen</b>	<b>Status</b>
Kollisionen: Fährt der virtuelle Roboter gegen eine wand, beginnt das System zu schwingen, d.h. die Zustände der Sensoren wechseln permanent, zwischen aktiviert und nicht aktiviert.	Da die Kollisionen korrekt gemeldet wird, haben wir einen Mechanismus eingebaut der testet, ob innerhalb eines gewissen Intervalls eine Kollision gemeldet wird. Solange mindestens eine Kollision im Iterval auftritt gilt der Sensor als aktiviert	Akzeptabler „workaround“: Die beste Lösung (bewegliche Roboter Fühler, auslenkung) kann im Rahmen der Diplomarbeit realisiert werden.
wenige Tests, keine Software tests mit einem Testing Framework	Die Applikation im Benutzer-Modus getestet	Uns sind keine gravierenden Probleme bekannt.
Die Laufzeitumgebung für LeJos-Programme kann unter windows nicht mit visual studio kompiliert werden.	Kompilieren der Laufzeitumgebung mit Cygwin	Gelöst, es konnte fehlerfrei kompiliert werden
Die Bibliothek der mit Cygwin kompilierten Laufzeitumgebung, konnte nicht mit MS Visual Studio eingebunden werden.	Alle Cygwin spezifische Funktionen wurden durch die entsprechenden Funktionen aus dem Wiindows-API ersetzt und die Laufzeitumgebung wurde im Windows-Kompatibilitäts-Modus mit Cygwin kompiliert.	Gelöst, die so generierte Bibliothek kann mit visual studio benutzt werden.



## 3.2 Ungelöste Probleme

Problem	Geetroffene Massnahmen	Status
Das Rendering-System von Ogre produziert je nach DirectX Version von Windows verschiedene Resultate.	Wir versuchten die Probleme so gut wie möglich zu umgehen (reine Darstellungs Probleme)	Nicht gelöst, wir hoffen, dass diese bekannten Probleme vom Ogre Entwicklungsteam bis im Herbst gelöst oder zumindest entschärft werden.
Die Crossplattform GUI-Frameworks wxWindows, Fltk, CGui und Gtk funktionieren nicht zufriedenstellend mit Ogre.	1. Verzicht auf ein GUI-Framework: Alle zusätzlichen Informationen für den Benutzer werden direkt in die 3D-Szene eingeblendet. 2. Die Auswahl des auszuführenden LeJos-Programms erfolgt über ein Kommandozeilen-Argument	Für einen Prototypen eine akzeptable Lösung, die aber im Rahmen der Diplomarbeit verbessert werden muss.
Unter MacOSX gab es Schwierigkeiten Boost (Threading Library) einzubinden	Da MacOSX keine hohe Priorität hat, haben wir diese Problem nicht angegangen	Nicht gelöst

## 4 Zielerreichung

Nachfolgende Übersicht listet die Ziele gemäss Projekthandbuch mit dem entsprechenden Resultaten auf:

Nr.	Ziel	Priorität	Erfüllt?
1	<b>Eine 3D-welt</b> Labyrinth	Mussziel	Ja
2	<b>Ein Lego Roboter Modell</b> Virtueller Roboter, der sich in der Landschaft bewegen kann	Mussziel	Ja
3	<b>Lauffähig mit leJOS &amp; Einbindung eigener leJOS Programme möglich</b> Eine Schnittstelle zwischen dem Lego Roboter Programmen und der 3D Landschaft. Über sie können die Roboter gesteuert und Sensor-Daten übermittelt werden	Mussziel	Ja

Nr.	Ziel	Priorität	Erfüllt?
4	<b>Basisinteraktion des Roboters mit der Umgebung</b> Hindernisse und Höhenunterschiede werden vom Roboter wahrgenommen und werden nach den allgemeingültigen physikalischen Gesetzen der Natur auf den Roboter einwirken	Mussziel	Ja
5	<b>Lauffähig auf Windows</b>	Mussziel	Ja
6	<b>Erweiterbare Architektur</b> Modularer Aufbau um eine möglichst einfache Weiterführung des Projekts in der Diplomarbeit zu ermöglichen	Kannziel	Ja
7	Lauffähig auf Windows, Linux	Kannziel	Ja
8	Mehrere Roboter gleichzeitig simulieren	Kannziel	Nein
9	Editor für Roboter	Kannziel	Nein
10	Zusätzliche Sensoren (Licht)	Kannziel	Nein
11	Emulator einsetzbar mit mehreren Lego-OS	Kannziel	Nein
12	Importer / Converter für Welten aus 3D Programmen	Kannziel	Nein

Die relativ hochgesteckten Ziele wurden alle erfüllt. Ebenfalls konnten wir mehrere Kannziele erreichen, so kann beispielsweise JORGE nicht nur auf Windows sondern auch auf Linux und teils auf MacOSX kompiliert und ausgeführt werden.

## 5 Gewonnene Erkenntnisse

### 5.1 Generell

Bei der Projekt-Initialisierung haben wir die anstehenden Aufgaben gut erkannt, so dass wir keine grundlegenden Korrekturen am Lösungsweg vornehmen mussten. Es hat sich auch gezeigt, dass die geplante Lösung machbar ist und gut funktioniert.

### 5.2 Zum Modell Hermes

Die verschiedenen Dokumente die das Modell Hermes fordert, hat zu einem Mehraufwand geführt, dessen Aufwand sich erst am Ende des Projekts ausgezahlt hat: Dank der ausführlichen und guten Planung können wir unser Projekt nun erfolgreich abschließen. Somit ist Hermes eine gute Richtlinie die einem hilft den roten Faden einzuhalten, manchmal muss man aber auch den Mut haben von den Forderungen von Hermes abzuweichen.



## 5.3 Technologie

Allgemein kann gesagt werden, dass die Entwicklung mit C++ sicherlich sehr viel aufwendiger ist als beispielsweise mit Java. Die Entwicklung auf 3 Betriebssystemen (Windows, Linux und MacOSX) hat zusätzlich zu Aufwand geführt.

### 5.3.1 Ogre

Für die Realisation des graphischen Teils haben wir Ogre gewählt. Als positiv zu bewerten sind vor allem die klare Strukturierung und die grosse Community.

### 5.3.2 Ode

Ode ist ein Physics Framework. Entscheidend für diese Wahl war hauptsächlich das bereits vorhandene Zusammenspiel zwischen Ogre und Ode. Dieses wurde über einen Wrapper realisiert, der sich bereits bewährt hat.

### 5.3.3 Blender

Um die 3D Objekte, wie beispielsweise unseren Roboter, zu kreieren haben wir auf OpenSource gesetzt. Blender ist auf allen von uns avisierten Betriebssystemen lauffähig und verfügt ebenfalls über eine grosse Community.

### 5.3.4 Emulator

Die Evaluation der verschiedenen Open Source Projekte war sicher interessant, schlussendlich aber auch ernüchternd, da von den acht getesteten Lösungen nur zwei in die engere Auswahl kamen. Das häufigste Problem der verschiedenen Projekte war die mangelnde Dokumentation, so dass der Source-Code analysiert werden musste, um eine Aussage über den Funktionsumfang zu machen.

## 5.4 Infrastruktur

Die uns zur Verfügung gestellte Hardware war sehr leistungsfähig und in sehr gutem Zustand. Dies hat es uns ermöglicht auch aufwändige Anwendung mühelos auszuführen, was gerade im Bereich der Computer Graphik sehr wichtig ist und entscheidend zum Endresultat beiträgt.



Die momentanen Platzverhältnisse sind überdurchschnittlich. So hatten wir unser eigenes „Büro“, in dem wir gemeinsam arbeiten konnten. Nicht optimal war die Heizung. Selbst bei schönem Wetter wurde die Heizung nicht ausgeschaltet, dadurch erreichte die Zimmertemperatur Höchstwerte.

## 5.5 Organisation

Die Zusammenarbeit mit unseren Fachdozent und unserem PM-Coach hat eine gute Basis geschaffen um unsere Ideen verwirklichen zu können. Durch regelmässige Sitzungen mit unseren Fachdozenten behielten wir stets den Überblick über unser Projekt. Durch eine klare Organisation, welche nicht zuletzt durch Planung errichtet wurde, erhielten wir die Sicherheit und das Selbstvertrauen um auch in schwierigen Situation ein Lösung zu finden.

## 5.6 Projektarbeit/Teamarbeit

Die Aufteilung der Arbeit in zwei Teile, 3D Welt und Laufzeitumgebung für LeJos-Programme, und die spät angesetzte Zusammenführung hat sich bewährt. Die gegenseitigen Abhängigkeiten waren sehr gering, so dass neue Funktionen sofort getestet werden konnten, und sich Verzögerungen der einen Seite nicht auf die andere Seite ausgewirkt haben.

Obwohl wir die Arbeiten aufgeteilt hatten, haben wir uns immer wieder gegesiegt geholfen Schwierigkeit zu besprechen und diese dann auch zu lösen.

## 5.7 Betreuung Fachdozent + PM Coach

Unsere Fachdozenten Herr Fuhrer und Herr Dubois haben uns immer wieder interessante und aufschlussreiche Inputs gegeben. Dank diesen Inputs konnten wir sehr viel Zeit sparen und es half uns den richtigen weg einzuschlagen. Dank der Hilfe von Herrn Helbling, unserem PM-Coach, konnten wir ausserdem ein gutes Fundament für unsere Arbeit legen, da die klare Planung und Dokumentierung entscheidend zur Erkennung und Fokussierung der Probleme beitrug.

## 6 Folgerungen für die Diplomarbeit/künftige Projektarbeit

Glücklicherweise haben wir in dieser ersten Phase keine bösen Überraschungen erlebt. Die von uns evaluierten Komponenten entsprachen voll unseren Erwartungen. So können wir von unserem erarbeiteten Wissen sicherlich während der Diplomarbeit profitieren. Ziel wird es sein, die Applikation zu optimieren und natürlich zu erweitern.

Ein weiterer Vorteil ist das Arbeitspensum in OpenSource Projekten. Wir können davon ausgehen, dass während der Sommerpause bereits wieder Verbesserungen und Erweiterungen an den Frameworks, wie Ogre, vorgenommen worden sind.

Da wir in der Projektarbeit bereits relativ weit gekommen sind, haben sich die Ziele für die Diplomarbeit ein wenig verändert. Wir wissen heute noch nicht genau, welche Ziele wir während der Diplomarbeit realisieren wollen. Wir werden aber sicherlich die Sommerpause dazu nutzen um uns die eine oder andere Idee durch den Kopf gehen zu lassen.

## 7 Persönliche Meinungen

### 7.1 Christof Seiler

Mit diesem Projekt haben wir nicht gerade den Weg des geringsten Widerstandes gewählt. Es war mit viel Aufwand verbunden, hat mir aber auch neue Einblicke und Einsichten ermöglicht. Es war sehr interessant sich für einmal nicht nur als Einzelkämpfer zu bewähren, sondern sich in ein Team einzufügen. Ich musste sogar mit Schrecken feststellen, dass nicht immer alle meiner Meinung waren, und es am Schluss darum ging einen geeigneten Kompromis zu finden.

Ich freue mich auf die Weiterführung dieses Projekts als Diplomarbeit und bin überzeugt davon erfolgreich mit meinen Teampartnern neue Aufgaben in Angriff zu nehmen.

### 7.2 Niklaus Lutz

Mit dieser Semesterarbeit habe ich viele wertvolle Erfahrungen gemacht, auf die ich später im Berufsleben zurückgreifen kann. Einmal mehr hat es sich gezeigt, dass die Kommunikation (innerhalb des Teams, aber auch mit den Betreuern) zusammen mit einer guten Planung wichtiger für den Erfolg des Projektes ist, als die reine Entwicklungsarbeit. Als wir uns bei Projektbeginn für C++ als Programmiersprache



entschieden hatten, befürchtete ich, dass das Projekt möglicherweise wegen dieser Entscheidung scheitern könnte, da ich nur den zusätzlichen Aufwand sah. Dieser Aufwand hat sich aber sehr gelohnt, ich glaube während dieser Semesterarbeit mehr C/C++ gelernt zu haben als in allen zuvor besuchten C/C++ Kursen.

### 7.3 Stefan Feissli

Für mich war es sehr interessant mal ein Projekt mit C++ zu realisieren. Auch wenn es nicht immer lustig war, denke ich, dass mir die Entwicklung unter Linux sehr viel gebracht hat. Was die Projektführung anbelangt, bin ich überzeugt, dass dieser praktische Weg sehr viel mehr bringt als eine Probe über den Stoff zu schreiben. Die Teamarbeit hat wie erwartet, die meiste Zeit Spass gemacht. Kurz gesagt, das Projekt war aufwändig, aber sicherlich sehr lehrreich und interessant.

## 8 Schlussfolgerungen

### 8.1 was lief gut

- Es war uns meist möglich komplexe Teilprobleme in kurzer Zeit zu lösen.
- Mit dem Schlussresultat sind wir sehr zufrieden und können uns gegenseitig auf die Schulterklopfen.
- Ein Punkt den wir immer gerne hervorheben ist unser Auftritt, mit Logo, Webpage und dem Design unserer Arbeit im Allgemeinen. Dank manchmal auch glücklichen Geschehnissen konnten wir eine stylistisch, einwandfreie Präsentation unseres Produktes ermöglichen.

### 8.2 was sollte beim nächsten Mal besser gemacht werden

- Es ist praktisch unmöglich in einem Dreierteam immer gleicher Meinung zu sein. Es brauchte zum Teil lange und ausführliche Diskussionen um auf einen gemeinsamen Nenner zu kommen. Das braucht Kraft und Nerven und ist nicht immer einfach. Der Informationsfluss innerhalb des Team sehr wichtig, welcher bei uns zum Teil nicht ideal war. Für spätere Projekte, im besonderen der Diplomarbeit, werden wir versuchen diesen Fluss der Informationen stärker zu berücksichtigen um auf Probleme frühzeitig reagieren zu können und lange, unproduktive Gespräche zu vermeiden.
- Für die Erledigung der meisten Aufgaben setzten wir auf Open Source Software. Dies ist sicherlich eine gute Sache, jedoch wollen wir in Zukunft darauf achten keine Beta Versionen mehr zu gebrauchen, da diese immer wieder grosse Überraschungen mit sich bringen.