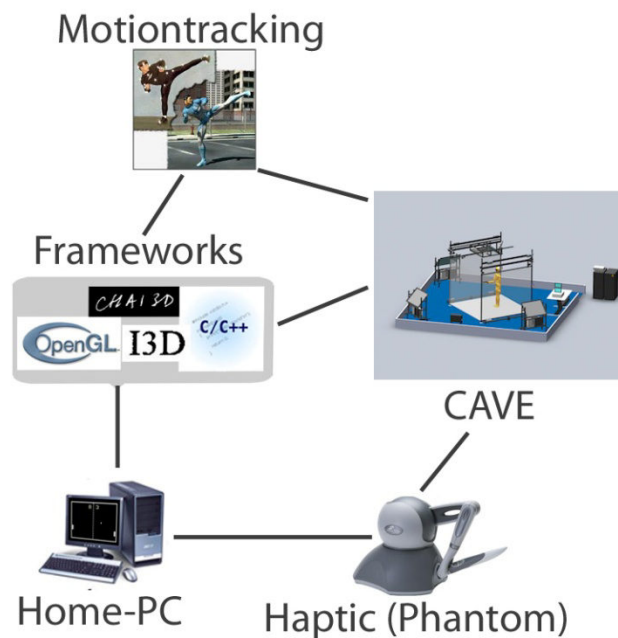


Bachelor Thesis 2012

SquashI3D

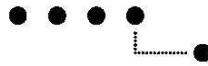
Pflichtenheft – V1.10



Fachbereich
Studierende

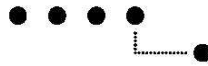
Professoren

BFH-TI Informatik
Andreas Emch
Daniel Pfäffli
Prof. Urs Künzler

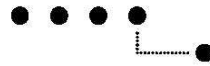


Inhaltsverzeichnis

Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	V
1 Einleitung	1
1.1 Zweck und Ziel dieses Dokuments	1
1.2 Projektstammdaten	1
2 Vorarbeit Projekt 2	2
3 Bachelor Thesis	4
4 Das Spiel Squash	5
4.1 Regeln	5
4.1.1 Aufschlag	5
4.1.2 Ballwechsel	6
4.1.3 Zählweise	6
4.2 Regeln im Squash3D	6
5 Framework - Übersicht	7
6 Anforderungen	8
6.1 Projektziele	8
6.2 Systemumgebung	9
6.3 Technical Requirements.....	10
6.3.1 Framework I3D.....	10
6.3.2 Level-Design mit OpenSceneGraph	10
6.3.3 Spiel - Client-Server Architektur	10
6.3.4 Input - Schnittstelle Haptik.....	11
6.3.5 Input - Motion Tracking.....	11
6.3.6 Spiel - 2-Spieler mit 2 Eingabegeräten	11
6.4 Visual Design	12
6.4.1 Design Basic	12
6.4.2 Modellierung.....	12
6.4.3 Rendering Basic.....	13
6.4.4 Rendering Advanced.....	13
6.5 Functional Requirements	13
6.5.1 Spiel - Multiplayer	13
6.5.2 Spiel - Singleplayer	14
6.5.3 Input - Geräte-Eigenschaften.....	14
6.5.3.1 Phantom	14
6.5.3.2 Inca 6D	14
6.5.4 Spielprinzip - Normales Squash	14
6.5.5 Spielprinzip - Verschiedene Räume.....	15
6.5.6 Spielprinzip - Squash als Break-Ball.....	15
6.5.7 Collisiondetection	15
6.5.8 Bewegungsfreiheiten des Spielers 1D.....	15
6.5.9 Bewegungsfreiheiten des Spielers 2D.....	15
6.5.10 Input - Haptisches Feedback	16
6.5.11 Input - Interaktion Haptic	16

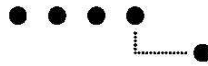


6.5.11.1 Phantom	16
6.5.11.2 Inca 6D	16
6.5.12 Input - Interaktion MotionTracking	16
6.5.13 Audio	16
6.5.14 Menueführung	17
6.6 Varianten.....	17
6.6.1 Varianten Input (VI)	17
6.6.2 Varianten Spiel (VS).....	18
6.6.3 Varianten Spielprinzip (VSP)	18
6.6.4 Vergleichskriterien.....	19
6.7 Fehlerbehebungen	21
6.8 Anforderung an systemexterne Schnittstellen	21
6.9 Anforderungen an systeminterne Schnittstellen.....	21
6.10 Anforderungen an die Systemumgebung	21
6.11 Priorisierung	22
6.12 Use-Cases	24
6.12.1 Spiel starten	25
6.12.2 Ball-Zustand ändern	26
6.12.3 Ball schlagen.....	26
6.12.4 Punkte addieren	28
7 Testing	29
8 Abgrenzung	29
9 Projektmanagement.....	30
9.1 Einleitung	30
9.2 Organigramm	30
9.3 Ressourcen.....	31
9.4 Projektvorgehen.....	31
9.5 Übersicht der Meilensteine.....	33
9.5.1 Pflichtenheft	33
9.5.2 Machbarkeitstest Haptic	33
9.5.3 Machbarkeitstest 2-Haptic Geräte	34
9.5.4 Verhaltenstest INCA 6D	34
9.5.5 MotionTracking Test.....	34
9.5.6 Software Design.....	34
9.5.7 Spieler-Bewegungen	34
9.5.8 Ball-Interaktion	35
9.5.9 Spielprinzip.....	35
9.5.10 Optimierungen.....	36
9.5.11 Endprodukt.....	36
9.6 Projektplanung	36
9.7 Projektorganisation	37
10 Anhang	39
10.1 Historie der Dokumentversionen	39



Abbildungsverzeichnis

Abbildung 1: Haptisches Gerät - Phantom Sensable.....	2
Abbildung 2: Haptic Pong.....	2
Abbildung 3: Spiel Squash	5
Abbildung 4: Spielfeld vom Squash	5
Abbildung 5: Framework – Übersicht	7
Abbildung 6: Systemübersicht.....	9
Abbildung 7: Modell-Prototyp	12
Abbildung 8: UseCases.....	24
Abbildung 9: UseCase – Spiel starten	25
Abbildung 10: UseCase – Ballmodus.....	26
Abbildung 12: UseCase – Punktegewinn.....	28
Abbildung 13: Organigram	30
Abbildung 14: Projektphasen	31
Abbildung 15: Gant-Diagramm –Teil 1	36
Abbildung 16: Gant-Diagramm –Teil 2.....	37
Abbildung 17: Aufbau SVN	37
Abbildung 18: Redmine	38



Tabellenverzeichnis

Tabelle 1: Projektstammdaten	1
Tabelle 2: Framework-Komponenten	7
Tabelle 3: Systemumgebung	10
Tabelle 4: Variantenentscheid	19
Tabelle 5: Legende Variantenentscheid	20
Tabelle 6: Priorisierung	23
Tabelle 7: Projektrollen	31
Tabelle 8: Projektphasen	32
Tabelle 9: Meilensteine	33
Tabelle 10: SVN	37
Tabelle 11: Versionierung	39



1 Einleitung

1.1 Zweck und Ziel dieses Dokuments

Dieses Dokument beschreibt die Rahmenbedingungen für und Anforderungen an das im Titel genannte Projekt. Es dient...

- zur Definition des Projektzieles
- des groben Projektverlaufs
- der Findung aller beteiligten Personen und Institutionen sowie der benötigten Ressourcen
- zur Vorbereitung eines Projektentscheids
- Kommunikation der Ziele und Meilensteine an alle Beteiligten

1.2 Projektstammdaten

Rolle / Rollen	Name	E-Mail
Dozent	Urs Künzler	urs.kuenzler@bfh.ch
Experte	Andreas Dürsteler	
Student	Andreas Emch	emcha1@bfh.ch
Student	Daniel Pfäffli	pfafd1@bfh.ch

Tabelle 1: Projektstammdaten



2 Vorarbeit Projekt 2

Im Rahmen des Projekt 2 haben wir das Spiel Pong auf eine haptische Ebene transformiert. Die Einschränkungen im haptischen Bereich waren dabei gegeben durch die Unterstützung von Freiheitsgraden des haptischen Gerätes.



Abbildung 1: Haptisches Gerät - Phantom Sensable

Optimiert und entwickelt wurde das Spiel für das haptische Gerät Phantom SensAble. Das Spiel Pong wurde um die 3. Dimension erweitert. Im Gegensatz zu bereits bestehenden 3D-Pong-Implementationen konnte ebenfalls die Steuerung in die 3. Dimension geführt werden. Durch die 3 Freiheitsgrade des haptischen Gerätes sieht der Spieler nicht nur die 3. Dimension, sondern erlebt diese auch. Die räumliche Begrenzung, der Aufprall des Balles/ Scheibe am Schläger/ Klotz, sowie der Klotz selbst geben dem Spieler über das haptische Gerät ein haptisches Feedback. Durch die Möglichkeit des Spürens der Aktionen erwarteten wir dabei eine höhere Immersion, als dies bei den herkömmlichen Pongs der Fall ist.

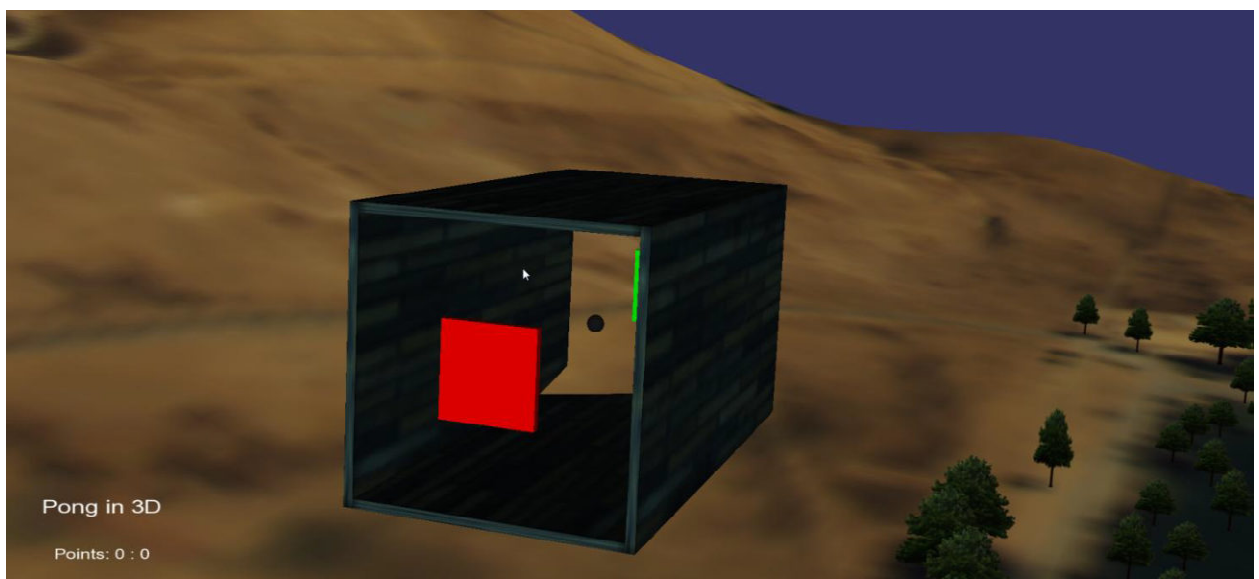


Abbildung 2: Haptic Pong



Wir haben uns sehr intensiv mit der gesamten Programmierung von I3D befasst. Hier haben wir vor allem die Möglichkeiten von dem OSG ausgelotet, was uns für eine effiziente Gestaltung des Levels ermöglicht. Hier haben wir viele Techniken wiedererkannt, welche wir schon im Kurs CPVR 2 kennen gelernt haben. In unserem Level von dem Projekt 2 sind einige Techniken wieder zu erkennen. So haben wir den Himmel als einfache Box mit einer Texturierung modelliert. Um einen fließenden Übergang zu erreichen, haben wir entsprechend Fog über den Rest des Levels gelegt. Der OSG-Baum ist so klein wie möglich gehalten, was in unserem Level nicht schwierig war. Trotzdem haben wir viel Zeit investiert, bis wir mit dem OSG schnell zu dem Resultat kamen, welches wir uns vorgestellt haben.

Mit der Haptic haben wir uns beschäftigt, als der Level fertig gestaltet war. Schnell haben wir erreicht, dass die Kugel als Box dargestellt wird. Jedoch mussten wir feststellen, dass die Implementierung im I3D für die Erweiterung eines neuen Pointers mühsam ist. Die HapticBulletSpherePointer war an vielen Stellen fest im Code integriert, was es schwieriger machte, diesen Pointer zu ersetzen. Aus diesem Grunde haben wir uns dazu entschieden, die relevanten Haptic-Klassen in unser Projekt auszulagern und dort anzupassen. So müssen wir nicht viel im I3D selbst ändern. Die Kollisionserkennung funktioniert leider noch nicht ganz optimal, ist aber schon besser als anfangs. Für die Bachelorthesis müssen wir uns noch einmal überlegen, wie wir diese Haptic-Implementierung umsetzen wollen in dem Projekt. Die Pointer sollten dynamisch geladen und austauschbar sein. Der andere Weg macht es zu schwierig, wenn alles statisch eingetragen ist, um einen neuen Pointer zu erstellen und anzeigen zu lassen.

Mit dem Bullet konnten wir uns leider nicht so intensiv beschäftigen, wie wir uns dies vorgestellt hatten. Wir mussten einige Zeit mehr für die Haptic aufwenden als zuerst noch angenommen. So haben wir uns auf das Minimale beschränkt mit dem Bullet. Der Ball bekommt eine Anfangsgeschwindigkeit und prallt entsprechend von den Wänden ab. Um die Bewegung noch zu beeinflussen, müssten die Panels mit einer Struktur versehen werden, damit der Ball auch ein bisschen gesteuert werden kann. In der Bachelorthesis müssen wir uns anfangs noch ein wenig in das Bullet vertiefen, damit wir dies ebenfalls optimal ausnutzen können.

Insgesamt haben wir uns viel mit dem I3D beschäftigt und wissen inzwischen, wo wir etwas anpassen müssen und wie wir es anwenden müssen für unsere Bedürfnisse. Wir haben einige Hilfsklassen geschrieben, wie zum Beispiel für verschiedene Tastaturlayouts oder Effekte für die Programmierung mit dem OSG, welche unsere Arbeit für die Bachelorthesis vereinfachen sollten.



3 Bachelor Thesis

Die Bachelor-Thesis ist die finale Schlussarbeit für unser Bachelor-Studium. Aufgrund unseren Interessen und unserer Fachrichtung CPVR haben wir uns auf das Projekt geeignet, ein Haptic-Spiel zu erstellen. Um diese Herausforderung zu bewältigen, haben wir das Projekt 2 so gewählt, dass wir uns mit der gesamten Thematik vertraut machen konnten. Wir haben bereits ein kleines Spiel mit dem I3D-Framework implementiert, welches auf das haptische Gerät reagierte. So lernten wir ebenfalls alle integrierten Frameworks kennen.

Unsere Bachelor-Thesis besteht aus einem weiteren 3D-Spiel, welches auf dem Projekt 2 aufbaut. Wir haben uns entschlossen, das Spiel Squash umzusetzen, welches im nächsten Kapitel genauer beschrieben wird. Das Spiel hat Ähnlichkeiten mit dem Spiel Pong auf, jedoch kommen weitere Elemente hinzu. So wird die Physik angepasst und das Model wird nicht mehr ein einfacher Block sein.



4 Das Spiel Squash

Mitte des 19. Jahrhunderts entstand das Spiel in England. In Deutschland wurden erst im Jahr 1930 die ersten Squash-Hallen gebaut. Das Spiel selber hatte aber dann doch erst ab 1970 den richtigen Aufschwung in Deutschland. Die Sportart ist aber bis heute noch keine Olympische Sportart.¹



Abbildung 3: Spiel Squash

Das Spielfeld ist definiert durch einen Raum mit einer Breite von 6.4 m, einer Länge von 9.75 m und einer Höhe von rund 4.75 m. Zusätzlich gibt es noch Auslinien, welche der Ball nicht überschreiten darf. Zusätzlich sind noch weitere Felder definiert wie zum Beispiel das Aufschlagfeld.

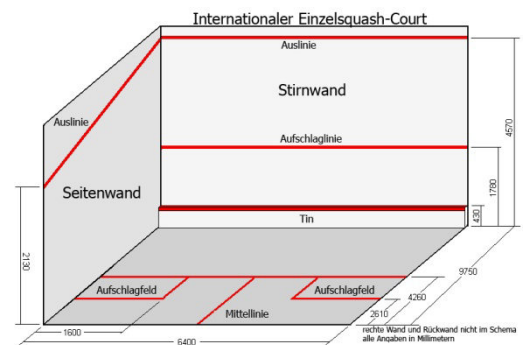


Abbildung 4: Spielfeld vom Squash²

4.1 Regeln

4.1.1 Aufschlag

Der Aufschlag wird immer aus einem der beiden Aufschlagfelder ausgeführt. Dabei muss der Aufschläger mit mindestens einem Fuß im Aufschlagfeld stehen. Nach dem Aufschlag muss der Ball die Stirnwand oberhalb der Aufschlaglinie treffen und auf der anderen Seite, im Viertel des Gegners, aufkommen. Beim ersten Aufschlag kann der Spieler die Aufschlagsseite frei wählen, muss dann aber die Seite nach jedem Punktgewinn wechseln. Wenn der Aufschläger den Ballwechsel verliert, erhält der Gegenspieler das Aufschlagrecht. Im Gegensatz zu anderen Racketsportarten darf der Ball beim Squash für den Aufschlag auch mit dem Schläger angeworfen werden.

¹ <http://de.wikipedia.org/wiki/Squash>

²

http://de.wikipedia.org/w/index.php?title=Datei:Squash_Court_German.jpg&filetimestamp=20081023112321



4.1.2 Ballwechsel

Der Ball muss nach jedem Schlag auf direktem oder indirektem Weg die Vorderwand berühren. Als indirekt gilt ein Weg über Seiten- und Rückwand. Danach darf der Ball nicht mehr als einmal auf dem Boden, jedoch beliebig oft auf die Rückwand oder Seitenwände auftreffen, bevor er vom Spielpartner zurückgeschlagen wird. Ein Ball gilt als im „Aus“, wenn er die Wände oberhalb dort angebrachter roter Begrenzungslinien, die Begrenzungslinie selbst oder das Tin berührt.

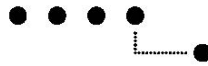
4.1.3 Zählweise

Es gilt, dass ein Spiel über drei Gewinnsätze geht, d. h. der Spieler, der als erster drei Sätze gewinnen kann, entscheidet das Spiel für sich. Es wird jeder Punkt gezählt, egal wer das Aufschlagrecht hatte. Für den normalen Satzgewinn benötigt der Spieler 11 Punkte. Beim Stand von 10:10 wird ein Tie-Break gespielt. Hier gewinnt derjenige Spieler, welcher zuerst 2 Punkte Vorsprung hat (z. B. 13:11, 19:17 usw.).

4.2 Regeln im Squash3D

Die Regeln werden wir in unserem Spiel, wie folgt, berücksichtigen:

- Der Ball muss nach jedem Schlag auf direktem oder indirektem Weg die Vorderwand berühren. Als indirekt gilt ein Weg über Seiten- und Rückwand.
- Der Ball darf auf dem Weg von der Vorderwand zum Spieler nicht mehr als einmal den Boden berühren.
- Ein Ball gilt als im „Aus“, wenn er die Wände oberhalb dort angebrachter roter Begrenzungslinien, die Begrenzungslinie selbst oder das Tin berührt.
- Die Rückwand gilt als „Aus“.



5 Framework - Übersicht

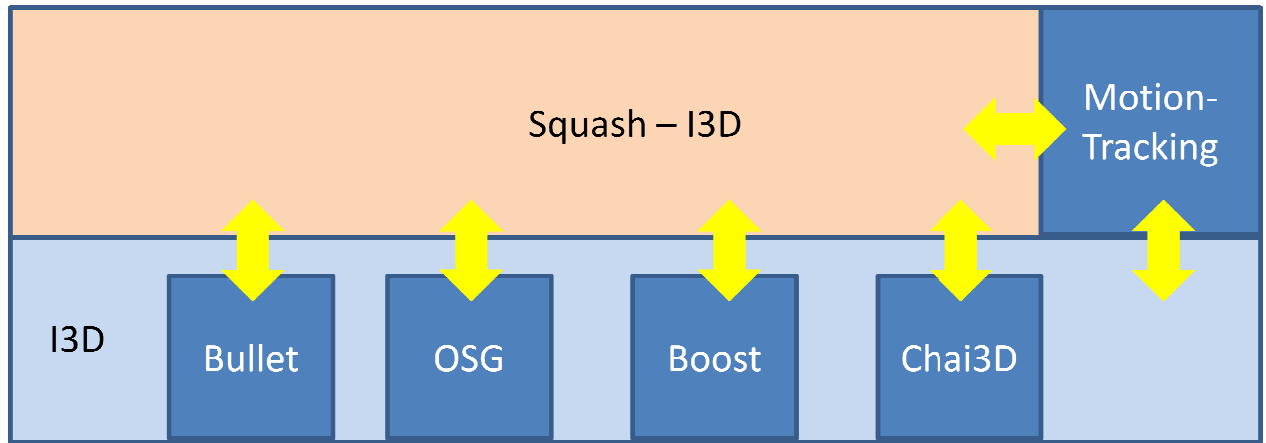


Abbildung 5: Framework – Übersicht

I3D – Komponenten	Beschreibung
Bullet	ist eine Physik-Engine, welche im I3D-Framework integriert wurde.
OSG	OpenSceneGraph erlaubt das Aufbauen einer 3D-Welt mit Hilfe eines Graphen. Dieses Framework wurde im I3D integriert.
Boost	umfasst viele Unterbibliotheken, welche Plattform-unabhängig verwendet werden können.
Chai3D	ist eine Haptic-Framework, welches ins I3D-Framework integriert wurde.
MotionTracking	bezeichnet ein Feature von I3D. Es erlaubt die Erkennung von Bewegungen im CAVE.

Tabelle 2: Framework-Komponenten



6 Anforderungen

Unser Spiel "Squash" muss einigen Anforderungen gerecht werden, welche im Folgenden genauer beschrieben sind.

6.1 Projektziele

Ziel der Projekt 2 Arbeit war es eine Basis zu schaffen, um mit Hilfe von haptischen Geräten ein Spiel zu konzipieren. In erster Linie ging es darum ein Spiel zu erstellen, bestehende Frameworks zu evaluieren und anzupassen, um mit Hilfe eines haptischen Gerätes Objekte im 3D-Raum zu ertasten und zu bewegen. Im Framework wird Wert gelegt auf die einzelne physikalische Gesetze (Fallgeschwindigkeit, Reibung, Trägheit, ...), sowie grundlegende Elemente des 3D-Renderings (Darstellung unter Wasser, Auseinanderbrechen von Materialien). Aufbauend auf den Resultaten aus der Projekt 2 Arbeit soll in dieser Bachelor-Arbeit ein haptisches Game designed und implementiert werden, welches über Netzwerk von zwei Playern gespielt werden kann und welches möglichst im CAVE lauffähig sein soll.

Als Spiel haben wir Squash als Vorbild genommen. Zusammengefasst können die Projektziele als folgende beschrieben werden.

Spielerlebnis im Bezug auf Squash:

- Squash ist im CAVE in einer Variante spielbar (siehe 6.6 Varianten)
- Im Spiel Squash werden die echten Spielregeln angewendet, jedoch wird der Realismus im Bereich Geschwindigkeit eingeschränkt.
- Die Bewegungen des menschlichen Spielerarms werden via Haptic oder MotionTracking im Spiel auf das Racket übertragen.

Technische Ziele

- Das Spiel läuft im CAVE
- Das Framework I3D kommt zum Einsatz



6.2 Systemumgebung

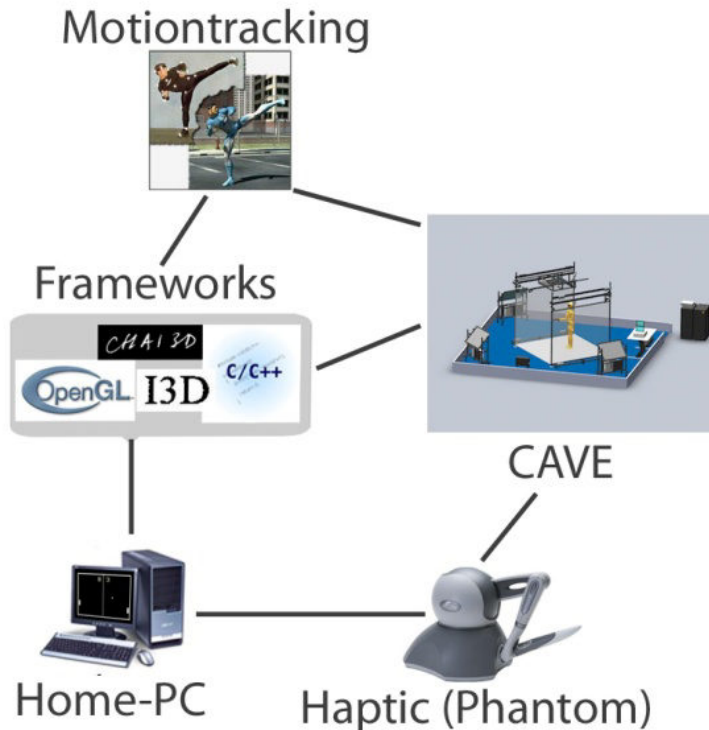


Abbildung 6: Systemübersicht

In der oben gezeigten Grafik sind unsere Hauptkomponenten aufgezeigt für unsere Projekt. Für die Hardware setzen wir den funktionierenden CAVE der BFH voraus. Nebst den üblichen Pheripherie-Geräten benötigen wir MotionTracking-Kameras.

Andere Software, welche wir benötigen ist das Visual Studio 2010 für die Programmierung in Visual C ++. Unser Projekt wird komplett in dieser Programmiersprache geschrieben. Als verschiedene Frameworks und Libraries verwenden wir das Framework I3D. Zusätzliche werden wir das Bildbearbeitungsprogramm „Adobe Photoshop CS“ verwenden für die Texturierungen, sowie das „Cinema 4D“ für die verschiedenen Modelierungen.

Die hier beschriebenen Komponenten dienen der Abgrenzung. Es werden nur die wichtigsten Teile der Hardware/ Software Umgebung genannt. Wir garantieren nicht dafür, dass auf anderen Umgebungen das Software-Verhalten dasselbe sein wird.



Entwicklungsumgebung

Typ	Hersteller
CPU	AMD Phenom II 955 BE, 64-Bit, Quadcore, 3.2 GHz
Grafikkarte	AMD HD 4890 AMD HD 7970
OS	Microsoft Windows 7
Entwicklungsumgebung	Microsoft Visual Studio 2010
Haptische Gerät	SensAble Phantom Omni Haptic Device

Tabella 3: Systemumgebung

Testsystem

Als Testsystem kommt der BFH-CAVE zum Einsatz. Stand: 28.09.2012

6.3 Technical Requirements

6.3.1 Framework I3D

Das I3D Framework ist von der BFH entwickelt worden um Anwendungen für den Cave zu ermöglichen. Das I3D funktioniert mit anderen bereits bestehenden Frameworks, so wie zum Beispiel mit Bullet und OpenSceneGraph. Ebenfalls ist das Chai3D und Boost darin implementiert.

Die Bachelor-Arbeit basiert als Endprodukt auf dem Framework I3D

6.3.2 Level-Design mit OpenSceneGraph

Unser Level wird durch den OpenSceneGraph (folgend OSG), welcher bereits im I3D integriert ist, gesteuert und verwaltet. Dieser SceneGraph wird verändert durch die Input-Geräte, sowie durch Bullet.

Das schlussendliche Spiel muss flüssig laufen und die Rechenzeit sollte nicht für unnötiges graphisches Rendering aufgebraucht werden. So muss der OSG geschickt aufgebaut werden und entsprechend optimiert werden.

Der Scenegrph umfasst, sowohl die Menüsteuerung, wie auch das eigentliche Spiel.

6.3.3 Spiel - Client-Server Architektur

Die Steuerung des Zweit-Spielers wird über den Input eines Netzwerk-Proxy gesteuert. Um dieses Ziel zu erreichen, setzen wir auf eine Client-Server Architektur. Die Kommunikation wird von 2 bis n Computern möglich sein, damit die Architektur problemlos für Spiele mit mehr als 2 Spieler eingesetzt werden kann. Die Kommunikation setzt das TCP voraus.

Im Bezug auf das Spiel Squash sehen die Anforderungen, wie folgt aus:

- Das System muss eine Server-Client Architektur beinhalten. Ein Spieler eröffnet das Spiel und definiert gegebenenfalls gewisse Einstellungen für das Spiel. Dieser Spieler wird automatisch zum Server des Spiels. Der zweite Spieler kann sich an dem Server anmelden und das Spiel kann gestartet werden.
- Die Kommunikation zwischen Server und Client wird mit TCP umgesetzt.



- Der Server stellt die zentrale Einheit des Spieles dar. Alle Clients senden die Nachrichten an den Server, welcher diese dann entsprechend verarbeitet und weiterverteilt.
- Die erhaltenen Nachrichten werden von jedem Client selber verwaltet (gespeichert für ein mögliches Rollback) und eine einfache Synchronisation erfolgt bereits auf dem Server sowie auf den einzelnen Clients.
- Unter Voraussetzung einer normalen Netzwerk-Architektur dürfen keine Verzögerungsbedingten Abbrüche erfolgen.

Die zu schickenden Informationen werden in einem Nachrichten Objekt verschickt. Dieses hat folgende Aufgaben:

- Es muss ein Timestamp enthalten für die Synchronisation der Spielzustände.
- Die nötigen Informationen für die optimale und schlanke Spielsynchronisation wird in der Nachricht enthalten sein.
- Änderungen des Spielzustandes müssen verschickt werden können. So kann eine Änderung sein, dass das Spiel startet, in den Pause-Modus wechseln usw.

6.3.4 Input - Schnittstelle Haptik

Als Haptisches Gerät kommt der SensAble Phantom Omni zum Einsatz. Das Spiel Squash wird auf dieses Gerät optimiert sein. Weiter wird das Spiel auf das INCA 6D optimiert. Andere haptische Geräte werden ebenfalls unterstützt, sofern die Schnittstelle die gleichen Funktionen bietet.

Die Ziele sind folgendermassen definierbar:

- Die Steuerung ist für das Phantom optimiert
- Die verschiedenen Freiheitsgrade müssen entsprechend freigeschalten oder eingeschränkt werden.
- Die entsprechenden Einschränkungen sind gegeben durch die Armlänge des 3D-Models. Die Arme dürfen wie bei einem realen Menschen nicht elastisch sein.
- Das Phantom Spielgerät muss dem Spieler ein haptisches Feedback zurückgeben.

6.3.5 Input - Motion Tracking

Motion Tracking wurde im Rahmen einer früheren Bachelor Arbeit im I3D integriert. Das Motion Tracking sollte mehrere Objekte gleichzeitig folgen können. Aufbauend auf Motion Tracking wird der Spieler mit Hilfe eines Kamera-Inputs gesteuert. Das MotionTracking-Geräte steuert dabei das Racket des Spielers.

Die Bewegungen des MotionTracking-Gerätes werden auf dem Racket übernommen.

6.3.6 Spiel - 2-Spieler mit 2 Eingabegeräten

Das Spiel wird auf dem gleichen Computer-System mit zwei Haptic-Devices ausgeführt. Die Anzahl Haptic-Devices generiert die Anzahl Spieler. Jedes Haptic-Device hat seinen eigenen Thread.

Diese Option stellt die Vorbedingung, dass das jetzige CAVE-System um einen weiteren Haptic-Computer erweitert werden kann. Weiter muss das Framework I3D die Information liefern, dass mehrere Haptic-Geräte im Einsatz stehen.



6.4 Visual Design

6.4.1 Design Basic

Die Modelle sind so modelliert, dass der Benutzer die Spielgegenstände erkennt. Der Spieler muss ebenfalls gut erkennen können, wo er im Raum steht und welche Positionierung das Racket hat.

Die Modelle sind mit Texturen versehen, um die Spielgegenstände auch graphisch ansprechend zu machen. Die Umgebung ausserhalb des Courts wird ebenfalls rudimentär gestalten werden.

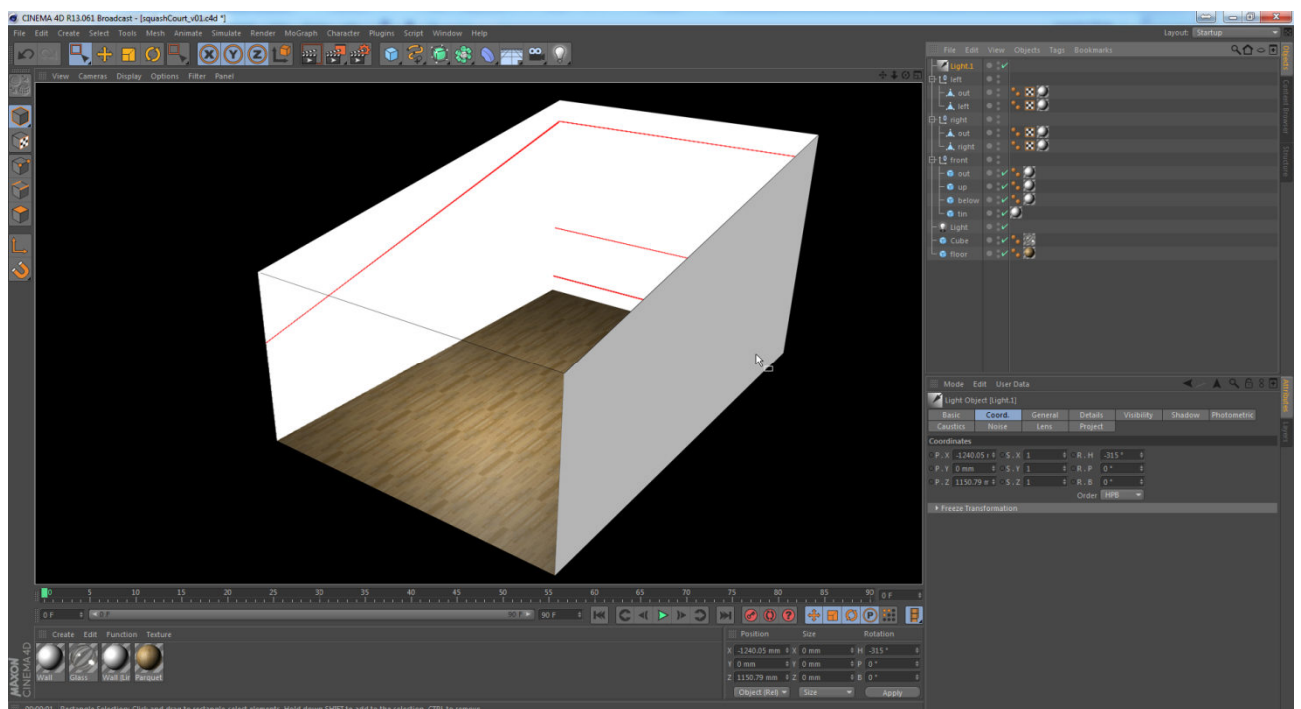


Abbildung 7: Modell-Prototyp

6.4.2 Modellierung

Die Modellierung wird einfach gehalten. Das Spielfeld selbst besteht aus Primitiven, welche mit Texturen versehen werden. Der Ball wird durch eine Sphere dargestellt. Einzig der Spieler selbst muss modelliert werden. Bei dem Spieler sind zusätzlich noch Animationen zu erstellen, welche aber wir jedoch einfach halten. Zusammengefasst haben wir folgende Modelle:

- Spieler mit Racket: Ein einfaches Modell eines Menschen mit einem Racket wird erstellt. Dazu sind die Bones festzulegen, welche die Animationen einfacher und dynamischer halten lassen.
- Squashball: Eine Kugel stellt den Spielball dar.
- Umgebung (Raum): Die Umgebung definiert sich durch den Raum selber. Dieser kann mit einfachen Blöcken zusammengesetzt werden. Die Blöcke sind so zu wählen, dass die Kollisionserkennung und Punktezahl optimal programmiert werden können.



- Umgebung (Halle): Die Umgebung ausserhalb des Raumes soll ebenfalls dargestellt werden. Dies kann mit einer einfachen Skybox realisiert werden, um mehr Detailgetreue zu erlangen.

6.4.3 Rendering Basic

Bei den Rendering-Funktionen werden wir unser Augenmerk auf die von OpenGL implementierten Techniken setzen. Dies Aufgrund der Zeitknappheit.

Das Squash-I3D wird nur wenige Lichtquellen erhalten, um die Komplexität gering zu halten.

Das System muss die Bewegungen in der 3.Dimension so darstellen, dass diese für den Spieler gut erkennbar sind. Der Spieler muss schnell begreifen, wie das Phantom mit seinem Racket zusammen agiert.

6.4.4 Rendering Advanced

Als Shading kommt ein Comic-Shader zum Einsatz, um das Squash-I3D in einem Comic-Stil zu halten.

6.5 Functional Requirements

6.5.1 Spiel - Multiplayer

Das Spiel wird einen Multiplayer-Modus zur Verfügung stellen. Dieser wird für genau 2 Spieler zur Verfügung stehen. Mehr Spieler für das gleiche Spiel können durch die Spielgegebenheiten nicht zugelassen werden. (Das Doppel im Squash lassen wir aus Zeitgründen aus)

Um gegeneinander zu spielen werden zwei Computer benötigt, welche über das Netzwerk oder mit zwei Eingabegeräten am gleichen Computer-System miteinander verbunden sind, um das Spiel gegeneinander anzutreten (Siehe 6.6 Varianten).

Zusätzlich zu den Regeln von 4.2 Regeln im SquashI3D kommen Folgende hinzu:

- Beim ersten Aufschlag wird rechts angefangen.
- Der Ball muss abwechselnd geschlagen werden.
- Weg Spieler 1 - Vorderwand: Berührt der Ball den Boden vor der Vorderwand, bekommt der Spieler 2 einen Punkt.
- Weg Spieler 1 - Vorderwand: Berührt der Ball den Spieler 2 vor der Vorderwand, bekommt der Spieler 1 einen Punkt.
- Weg Vorderwand - Spieler 2: Berührt der Ball zwei Mal den Boden vor dem Racket, bekommt Spieler 1 einen Punkt.
- Weg Vorderwand - Spieler 2: Berührt der Ball den Spieler 1 vor dem Racket von Spieler 2, bekommt Spieler 2 einen Punkt.
- Dieselben Regeln gelten für die entgegengesetzte Richtung.
- Für den normalen Satzgewinn benötigt der Spieler 11 Punkte.



6.5.2 Spiel - Singleplayer

Der Singleplayer-Modus sieht ein Spiel vor, in dem der Spieler mit sich selbst Squash spielt. Durch die Spielgegebenheiten ist der Singleplayer-Modus ohne KI zu bewältigen. Ziel ist ein möglichst langer Ballwechsel.

Zusätzlich zu den Regeln von 4.2 Regeln im SquashI3D kommen Folgende hinzu:

- Gezählt werden die Anzahl gültige Ballwechsel mit sich selber.

6.5.3 Input - Geräte-Eigenschaften

Für Haptic stehen zwei Geräte zur Auswahl. Je nach Priorität/ Zeit wird das eine oder/ und das andere eingesetzt.

6.5.3.1 Phantom

Der Spieler wird durch die Tastatur im Raum bewegt. Zusätzlich zu der Raumpositionierung wird mit dem Phantom die Positionierung des Rackets relativ zum Spieler gesteuert. Hier soll das Phantom blockieren, sobald der Schläger in eine unmögliche Position gebracht wird (z.B. wenn der Arm plötzlich länger werden müsste um die Positionierung einzunehmen). Weitere Einschränkung auf Trägheit oder weiteres werden nicht vorgenommen.

Die Kamera wird nicht neu positioniert im Verlaufe des Spiels. Der Spieler muss bereits mit der Tastatur und dem Phantom den Spieler steuern, was eine zusätzliche Funktion nur komplizierter macht.

6.5.3.2 Inca 6D

Das Geräte INCA 6D bietet 6 Freiheitsgrade. Diese Freiheitsgrade werden im Spiel je nach Variante ausgenutzt (siehe 6.6 Varianten). Mit Hilfe des INCA 6D wird die Raumpositionierung des Rackets relativ zum Spieler gesteuert. Hier soll das INCA 6D blockieren, sobald der Schläger in eine unmögliche Position gebracht wird (z.B. wenn der Arm plötzlich länger werden müsste um die Positionierung einzunehmen). Weitere Einschränkung auf Trägheit oder weiteres werden nicht vorgenommen.

6.5.4 Spielprinzip - Normales Squash

Das Level ist aufgebaut wie ein Squash-Court selber. Dieses ist definiert durch den Raum selber. Der Spieler kann sich nur in diesem Raum bewegen und muss sich an die gegebenen Spielregeln halten. Um den Ball wieder an die Front-Wand zu befördern, muss der dem Ball mit dem Racket einen Schwung mitgegeben werden, dieser wird durch die Bewegung des Phantoms berechnet. Das Phantom ist somit in alle Richtungen bedienbar und wie das Racket den Ball trifft wird dieser physikalisch korrekt abprallen. Die Spielregeln sind bereits unter dem Punkt 4.2 Regeln beschrieben.

Der Spieler steuert immer den rechten Arm des Modelles.



6.5.5 Spielprinzip - Verschiedene Räume

Aufbauend auf dem normalen Spielfeld (siehe 6.5.4) können dem Spieler verschiedene Räume mit unterschiedlichen physikalischen Eigenschaften zur Verfügung gestellt werden. Beispiele:

- Elastischer Raum
- Unterwasser Raum
- Raum ohne Gravitation
- Backstein Raum

Die Software-Architektur muss es zu lassen, dass einfach neue Räume durch Ableiten hinzugefügt werden können.

6.5.6 Spielprinzip - Squash als Break-Ball

Die Squash-Regeln (Siehe 4.1) werden beibehalten. Das Spielfeld wird erweitert um Klötze, welche durch anspielen verschwinden. Als Idee wird dabei das Spiel Break-Ball genommen.

6.5.7 Collisiondetection

Die Kollisionserkennung muss folgende Fälle abdecken:

- Racket zu Ball
Der Ball wird vom Racket getroffen => der Geschwindigkeitsvektor für den Ball muss neu gesetzt werden, abhängig von der Beschleunigung etc.
- Spieler zu Ball
Der Ball prallt an dem Spieler ab.
- Spieler/Racket zu Wand
Der Spieler kann nicht durch die Wand gehen. Ebenfalls soll das Racket nicht durch die Wand hindurch gehen und vom Phantom korrekt abgeblockt werden.
- Ball zu Wand
Der Ball prallt an eine Wand und muss physikalisch korrekt wieder abgeprallt werden.

Eine Kollision mit dem Spieler-Gerät muss auf dem Spielgerät wahrnehmbar sein (sofern dies unterstützt wird).

6.5.8 Bewegungsfreiheiten des Spielers 1D

Die Bewegungsfreiheiten der Spielerfigur werden auf eine Achse beschränkt. Das bedeutet, dass der Spieler sich nur auf einer Linie im Feld bewegt und von dort aus die Bälle schlagen muss. Die Rackets hingegen können im Raum bewegt werden.

Der Spieler kann sich seitwärts im Raum bewegen.

Wird dieses Feature umgesetzt, hat dies folgen auf die definierten Regeln/ UseCases:

- Der Ball muss direkt oder indirekt (max. 1 Bodenberührung) zur Linie der Spieler zurückkommen. Ist dies nicht der Fall, gilt der Punkt dem Spieler, der mit Schlägen an der Reihe wäre.

6.5.9 Bewegungsfreiheiten des Spielers 2D

Die Spielerfigur kann sich im 2D Raum bewegen. Sie kann sich sowohl Vorwärts/ Rückwärts/ Seitwärts bewegen.



6.5.10 Input - Haptisches Feedback

Das haptische Feedback besteht aus folgenden Fällen:

- Der Ball prallt am Racket ab, somit wird das Phantom/ INCA 6D einen kleinen Rückstoss simulieren.
- Das Racket befindet sich am Randbereich des möglichen Radius des Spielerarms. Falls die Position noch weiter geht wäre es unrealistisch, so muss das haptische Gerät dies verhindern.
- Falls der Spieler das Racket mit der Wand berührt, muss dies ebenfalls spürbar sein.

Die Rückmeldung an das Phantom ist soweit immer in Form eines konkreten Kraftvektors, welcher korrekt berechnet werden muss.

6.5.11 Input - Interaktion Haptic

Für die Interaktion mit Haptic stehen zwei Geräte zur Auswahl. Je nach Priorität/ Zeit wird das eine oder/ und das andere eingesetzt.

6.5.11.1 Phantom

Das SensAble Phantom Omni Gerät bietet 6 Freiheitsgrade, kann jedoch ein haptisches Feedback nur an 3 Freiheitsgrade zurück geben. Das Spiel Squash-I3D muss alle diese Freiheitsgrade benutzen. Der Spieler hat die Möglichkeit mit Hilfe des haptischen Gerätes sein Racket in die Tiefe/ Höhe/ Breite zu steuern. Zusätzlich wird noch umgesetzt, dass das Racket die Lage des Stiftes einnimmt, so hat der Spieler noch mehr Einfluss auf die Ballkontrolle.

6.5.11.2 Inca 6D

Zu den Freiheitgraden im SensAble Phantom Omni beschrieben, kommen drei Rotationsfreiheitsgrade. Diese sollen ebenfalls unterstützt werden.

6.5.12 Input - Interaktion MotionTracking

Das MotionTracking bewegt den Arm eines Spielers. Die Bewegungen des MotionTracking-Gerätes werden im Spiel ebenfalls so abgebildet. Das Racket folgt den Bewegungen des MotionTracking-Gerätes.

6.5.13 Audio

Das System muss eine Hintergrundmusik abspielen. Der Spieler muss die Möglichkeit haben, die Hintergrundmusik ein-/ ausschalten zu können.

Ebenfalls sollen gewisse Effekte hervorgehoben werden:

- Wenn eine Kollision geschieht, muss dies entsprechend per Ton dem Spieler angedeutet werden.
- Die Distanz der Kollision zum Spieler muss sich auf die Lautstärke auswirken.

Ebenfalls muss die Option wählbar sein, ob diese Sound-Effekte abgespielt werden oder nicht und dem Spieler steht eine Lautstärkeregelung zur Verfügung. So kann er die Musik und die Effekte separat anpassen.



6.5.14 Menuführung

Die Menuführung vor dem Spiel muss via dem Phantom, der Maus oder der Tastatur bedienbar sein. Die Umsetzung der Funktionen wird dabei mit Hilfe von vier Tasten gemacht. Zwei Tasten um sich durch das Menu zu bewegen und zwei Tasten um die Werte der Menu-Punkte zu verändern.

Das Menu muss folgende Optionen anbieten:

Startpunkt:

- Auswahl Spielermodi
- Ein-/ Ausschalten der Musik
- Spiel beenden

Zusätzliche Punkte je nach Variante:

- Singleplayer
- Multiplayer
- Spiel eröffnen => Server-Verbindungsdaten
- Spiel beitreten => Client-Verbindungsdaten

Während des Spieles:

- Ein-/ Ausschalten der Musik
- Zurück zum Startpunkt
- Spiel beenden

6.6 Varianten

6.6.1 Varianten Input (VI)

VI1: Motion Tracking

Der Input im CAVE wird via Motion-Tracking implementiert. Mit dieser Variante kann der Spieler ein Racket in den Händen halten und die Position und Lage des Rackets-im Spiel wird davon abgelesen. Ein haptisches Feedback ist in dieser Variante leider nicht möglich.

- 6.3.5 Input - Motion Tracking
- 6.5.12 Input - Interaktion MotionTracking

VI2: Haptic

Der Input im CAVE wird über ein haptisches Gerät (INCA 6 oder Phantom) eingelesen und entsprechend verarbeitet. In dieser Variante ist ein haptisches Feedback möglich, jedoch ist diese Variante abhängig von der Haptic-Integration im I3D.

- 6.3.4 Input - Schnittstelle Haptik
- 6.5.10 Input - Haptisches Feedback
- 6.5.11 Input - Interaktion Haptic
- 6.5.3 Input - Geräte-Eigenschaften



6.6.2 Varianten Spiel (VS)

VS1: Einzel

Das Spiel Squash wird als Einzelspieler gespielt. Hier ist das Ziel, einen möglichst langen Ballwechsel mit sich zu erreichen. Die Regeln des Spieles sind somit ein bisschen abgeändert, jedoch nicht was die Aufschlag-Regel und Out-Regel betreffen.

- 6.5.2 Spiel - Singleplayer

VS2: 2-Spieler mit Netzwerk

Zwei Spieler können in dieser Variante gegeneinander antreten. Hierfür benötigt jeder Spieler seinen eigenen Computer mit einem entsprechenden Eingabegerät. Die Synchronisierung des Spieles wird via Netzwerk ausgeführt. Diese Variante ist im CAVE so zu lösen, dass ein Spieler im CAVE spielt und der andere an einem anderen Computer im selben Netzwerk.

- 6.3.3 Spiel - Client-Server Architektur
- 6.5.1 Spiel - Multiplayer

VS3: 2-Spieler mit 2 Eingabegeräten

Zwei Spieler treten in einem Spiel lokal am gleichen Computer gegeneinander an. Die Synchronisation ist deshalb soweit zu ignorieren, weil beide am gleichen Spielstand agieren. Wie diese Variante am Besten im CAVE zu lösen ist, müssen wir zuerst eruieren. Je nach Eingabegeräte werden die Freiheiten des Spielers beschnitten werden müssen.

- 6.3.6 Spiel - 2-Spieler mit 2 Eingabegeräten
- 6.5.1 Spiel - Multiplayer

6.6.3 Varianten Spielprinzip (VSP)

VSP1: Normales Squash

Hier wird ein reelles Spielfeld kopiert. Der Parquet-Boden und die Wände mit den entsprechenden Massen sind korrekt auf das Modell angewendet. Dies ermöglicht es, das Squash reell zu simulieren.

- 6.5.4 Spielprinzip - Normales Squash

VSP2: Verschiedene Räume

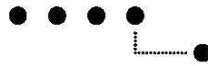
Das Spielfeld soll nicht nur dem von dem reellen kopiert werden, sondern es sollen neue Spielfelder implementiert werden können. Die Eigenschaften der Wände sowie das Visuelle können die Räume voneinander unterscheiden.

- 6.5.5 Spielprinzip - Verschiedene Räume

VSP3: Squash als Break-Ball

Das altbekannte Spiel Break-Ball soll in die dritte Dimension überführt werden. So muss der Spieler alle Blöcke vom Spielfeld räumen, ohne dabei einen Ballverlust zu erleiden.

- 6.5.6 Spielprinzip - Squash als Break-Ball



6.6.4 Vergleichskriterien

Mit Hilfe von verschiedenen Kriterien haben wir die verschiedenen Varianten beurteilt. Somit konnten wir uns besser entscheiden, wie wir die unterschiedlichen Varianten gewichten. Um eine bessere Beurteilung zu machen haben wir die Kriterien untereinander noch zusätzlich gewichtet. Die folgende Tabelle zeigt auf, wie wir die verschiedenen Varianten gewertet haben und was das berechnete Total der Werte ist.

Kriterium	Werte	Faktor (Gewichtung)	VI 1	VI 2	VS 1	VS 2	VS 3	VSP 1	VSP 2	VSP 3
Machbarkeit	1-5	5	4	3	5	3	2	5	5	5
Risiko	1-5	3	3	2	5	4	2	5	3	3
Geschätzter Aufwand	1-5	1	1	3	5	2	3	5	2	2
Know-How	1-5	1	1	4	5	3	4	5	4	4
Mehrwert	1-5	5	3	4	1	2	5	1	4	2
Total		Max. 75	46	48	55	42	48	55	60	50

Tabelle 4: Variantenentscheid

Somit haben wir folgende Varianten als höchste Prioritäten definiert:

- VI2
- VS1
- VSP2

Die anderen Varianten werden je nach Zeit ebenfalls implementiert.



Legende:

Kriterium	Von	Bis
Machbarkeit	1=Grosse Schwierigkeiten voraussehbar	5=Alle Schwierigkeiten bekannt und Lösungen sind vorhanden
Risiko	1=Viele unbekannte Gebiete, hohes Risiko, dass die Schätzung daneben ist.	5=Es gibt keine Risiken, alle Gebiete sind bekannt.
Geschätzter Aufwand	1=Voraussichtlicher Aufwand ist sehr gross	5=Voraussichtlicher Aufwand ist klein
Know-How	1=Der Bereich ist unbekannt, es braucht Einarbeitungszeit.	5=Es braucht keine Einarbeitungszeit, das Know-How ist vorhanden
Mehrwert	1=Die Variante bietet kaum einen Mehrwert für den Benutzer	5=Der Mehrwert für den Benutzer ist sehr gross.

Tabelle 5: Legende Variantenentscheid



6.7 Fehlerbehebungen

Mit der Haptic haben wir uns beschäftigt, als der Level fertig gestaltet war. Schnell haben wir erreicht, dass die Kugel als Box dargestellt wird. Jedoch mussten wir feststellen, dass die Implementierung im I3D für die Erweiterung eines neuen Pointers mühsam ist. Die HapticBulletSpherePointer war an vielen Stellen fest im Code integriert, was es schwieriger machte, diesen Pointer zu ersetzen. Aus diesem Grunde haben wir uns dazu entschieden, die relevanten Haptic-Klassen in unser Projekt auszulagern und dort anzupassen. So müssen wir nicht viel im I3D selbst ändern. Die Kollisionserkennung funktioniert leider noch nicht ganz optimal, ist aber schon besser als anfangs. Für die Bachelorthesis müssen wir uns noch einmal überlegen, wie wir diese Haptic-Implementierung umsetzen wollen in dem Projekt. Die Pointer sollten dynamisch geladen und austauschbar sein. Der andere Weg macht es zu schwierig, wenn alles statisch eingetragen ist, um einen neuen Pointer zu erstellen und anzeigen zu lassen.

6.8 Anforderung an systemexterne Schnittstellen

Die eingesetzte externe Schnittstelle Phantom SensAble, sowie die Cave-Geräte werden durch die Treiber der Hersteller abgedeckt. Das I3D-Framework, sowie das Chai3D Framework enthalten die Implementierungen der Schnittstellen bereits in ihren Frameworks.

6.9 Anforderungen an systeminterne Schnittstellen

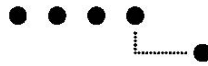
Die systeminternen Schnittstellen sind folgende Komponente:

- Haptic-Library/ MotionTracking-Library
- Boost (Network-Modul)
- Graphik-Umsetzung von OSG
- Bullet (Physik)

6.10 Anforderungen an die Systemumgebung

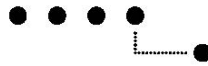
Die Entwicklung, sowie Testumgebung baut auf Microsoft Windows 7, 64-bit auf.

Die Entwicklungsumgebung basiert auf Prozessoren und Grafikkarten von AMD. Die CAVE-Hardware (Testsystem) ist mit Intel Prozessoren und Grafikkarten ausgerüstet. Allfällige unterschiedliche Verhalten der Software auf anderen Umgebungen, werden nicht getestet.



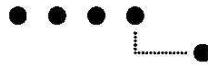
6.11 Priorisierung

Ziel		Prio 1 Muss	Prio 2 Soll	Prio 3 Soll	Meilenstein
Technical Requirements					
6.3.1	Framework I3D	✓			9.5.7 Spieler-Bewegungen
6.3.2	Level-Design mit OpenSceneGraph	✓			9.5.8 Ball-Interaktion
Varianten Input (VI)					
6.3.5	Input - Motion Tracking			✓	9.5.7 Spieler-Bewegungen
6.3.4	Input - Schnittstelle Haptik	✓			9.5.7 Spieler-Bewegungen
Varianten Spiel (VS)					
6.3.6	Spiel - 2-Spieler mit 2 Eingabegeräten		✓		9.5.9 Spielprinzip
6.3.3	Spiel - Client-Server Architektur			✓	9.5.9 Spielprinzip
Visual Design					
6.4.1	Design Basic	✓			9.5.7 Spieler-Bewegungen
6.4.2	Modellierung	✓			9.5.7 Spieler-Bewegungen
6.4.3	Rendering	✓			9.5.7 Spieler-Bewegungen
6.4.4	Rendering Advanced		✓		9.5.10 Optimierungen
Functional Requirements					
6.5.7	Collisiondetection	✓			9.5.8 Ball-Interaktion
6.5.14	Menüführung		✓		9.5.9 Spielprinzip
6.5.8	Bewegungsfreiheiten des Spielers 1D	✓			9.5.8 Ball-Interaktion
6.5.9	Bewegungsfreiheiten des Spielers 2D			✓	9.5.8 Ball-Interaktion
6.5.13	Audio		✓		9.5.9 Spielprinzip
Varianten Input (VI)					
6.5.10	Input - Haptisches Feedback	✓			9.5.7 Spieler-Bewegungen
6.5.11.1	Input - Interaktion Haptic Phantom	✓			9.5.7 Spieler-Bewegungen
6.5.11.2	Input - Interaktion Haptic Inca 6D		✓		9.5.7 Spieler-Bewegungen



6.5.3.1	Input - Geräte-Eigenschaften Phantom	✓			9.5.7 Spieler- Bewegungen
6.5.3.2	Input - Geräte-Eigenschaften Inca 6D		✓		9.5.7 Spieler- Bewegungen
6.5.12	Input - Interaktion MotionTracking			✓	9.5.7 Spieler- Bewegungen
Varianten Spiel (VS)					
6.5.2	Spiel - Singleplayer	✓			9.5.9 Spielprinzip
6.5.1	Spiel - Multiplayer		✓		9.5.9 Spielprinzip
Varianten Spielprinzip (VSP)					
6.5.5	Spielprinzip - Verschiedene Räume	✓			9.5.9 Spielprinzip
6.5.4	Spielprinzip - Normales Squash		✓		9.5.9 Spielprinzip
6.5.6	Spielprinzip - Squash als Break-Ball			✓	9.5.9 Spielprinzip
Fehlerbehebungen					
6.7	Fehlerbehebungen		✓		

Tabelle 6: Priorisierung



6.12 Use-Cases

UseCases sind für ein Spiel schwierig zu finden. Für Computerspiele erstellt man im Normalfall ein Storyboard. Anbei haben wir gewisse Use-Case-Szenarien entworfen, welche uns vor allem für die Testphase helfen werden. Die Szenarien sind wegen dem sehr zufälligen Spielverlauf generell gehalten.

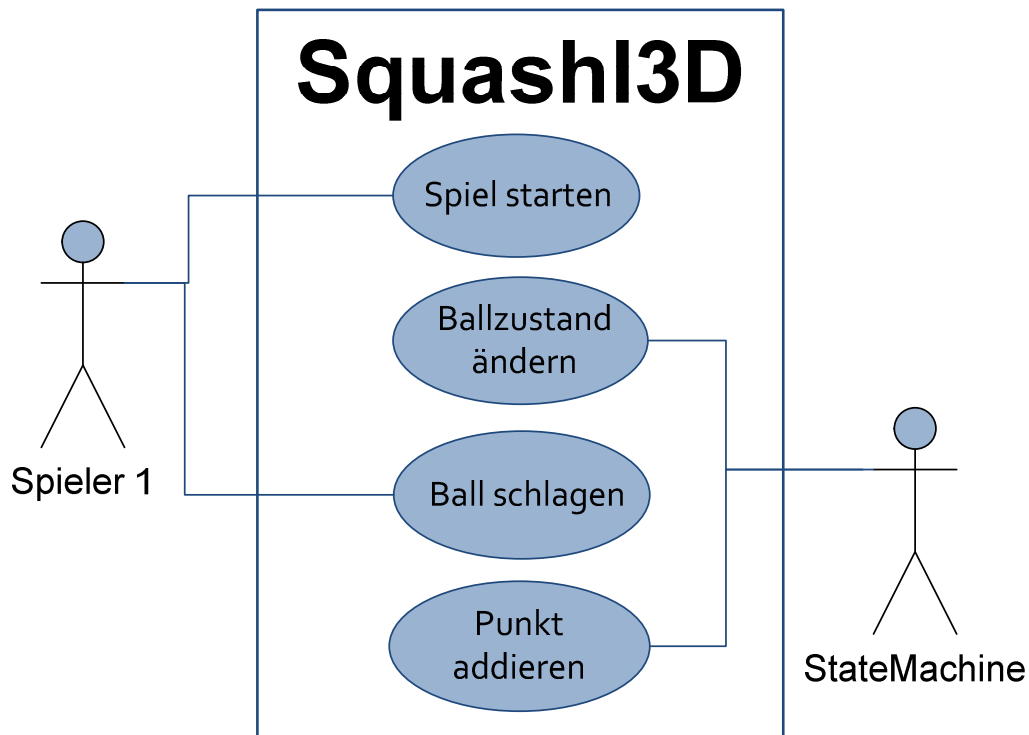


Abbildung 8: UseCases



6.12.1 Spiel starten

Nach dem Starten der BAT-Datei, welche zuständig ist, um die Anwendung im CAVE zu starten, hat der Spieler im CAVE die Möglichkeit, verschiedene Einstellungen gemäss 6.5.14 Menüführung zu ändern und/oder das Spiel zu starten.

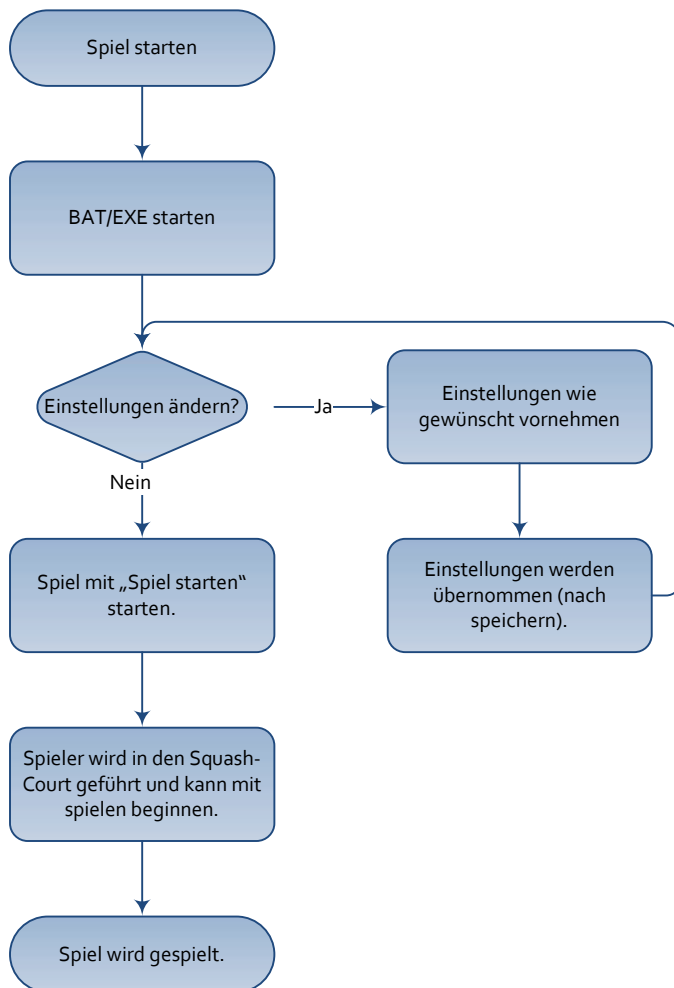


Abbildung 9: UseCase – Spiel starten



6.12.2 Ball-Zustand ändern

Der Aufschlag wird im Squash als einen Spezialfall von den üblichen Ballwechselln gesehen. Beim Aufschlag muss der Ball oberhalb der Aufschlaglinie der Frontwand abprallen. Ansonsten ist es ein ungültiger Aufschlag. Um dies möglichst einfach realisieren zu können, wird der Ball mit einem Modus-Attribut versehen, um beim Aufprall an der Frontwand den entsprechenden Fall überprüfen zu können.

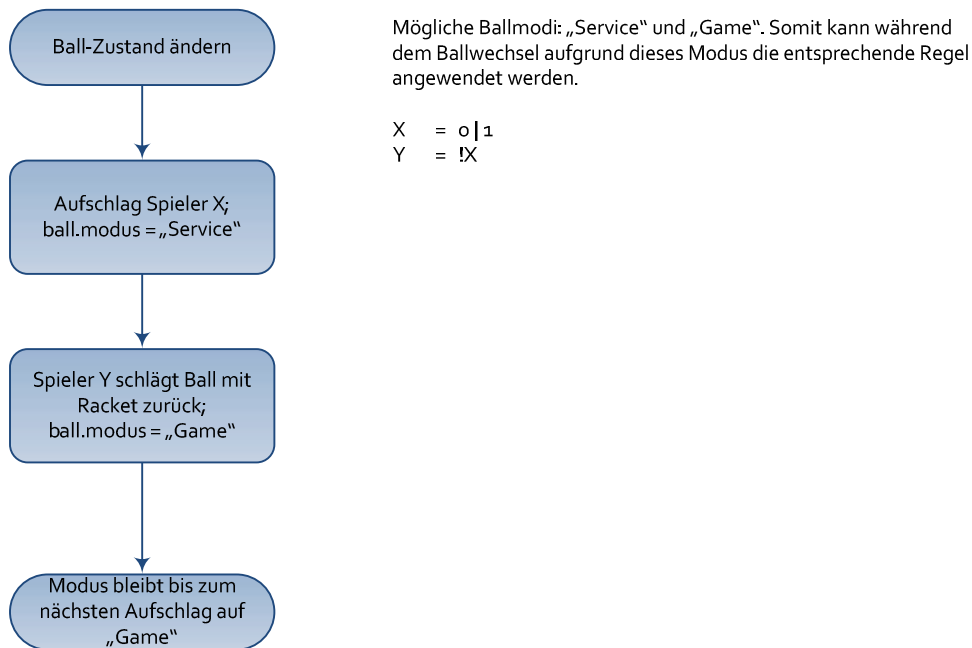


Abbildung 10: UseCase – Ballmodus

6.12.3 Ball schlagen

Der Ballwechsel betrifft folgende Zeitspanne: Spieler X schlägt mit dem Racket den Ball bis zum Zeitpunkt, wo der Spieler Y diesen mit seinem Racket abnimmt. Anschliessend beginnt ein neuer Ballwechsel.

Der Ballwechsel wird ausserdem beendet, sobald ein ungültiger Fall eintritt, welcher die Regeln bricht.

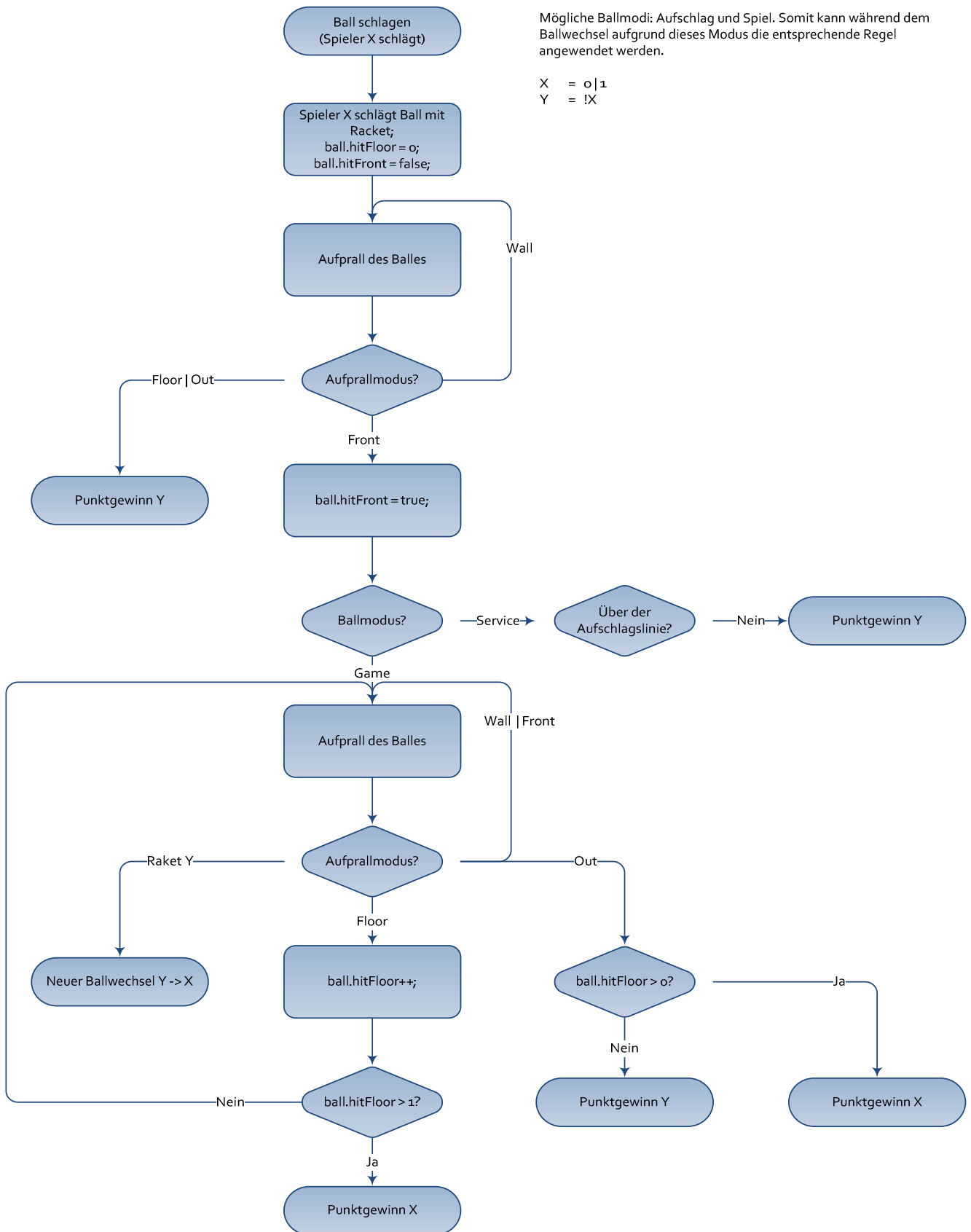
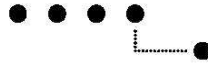


Abbildung 11: UseCase – Ballwechsel



6.12.4 Punkte addieren

Ziel eines jeden Spielers ist es, als Erster die benötigte Punktzahl zu erreichen. Der erste Spieler, welcher diese erreicht, gewinnt den Satz und in unserem Spiel auch gleich das gesamte Spiel. In diesem Use-Case wird definiert, wann ein solcher Gewinner festgelegt wird. Startpunkt des UseCases ist ein Punktegewinn.

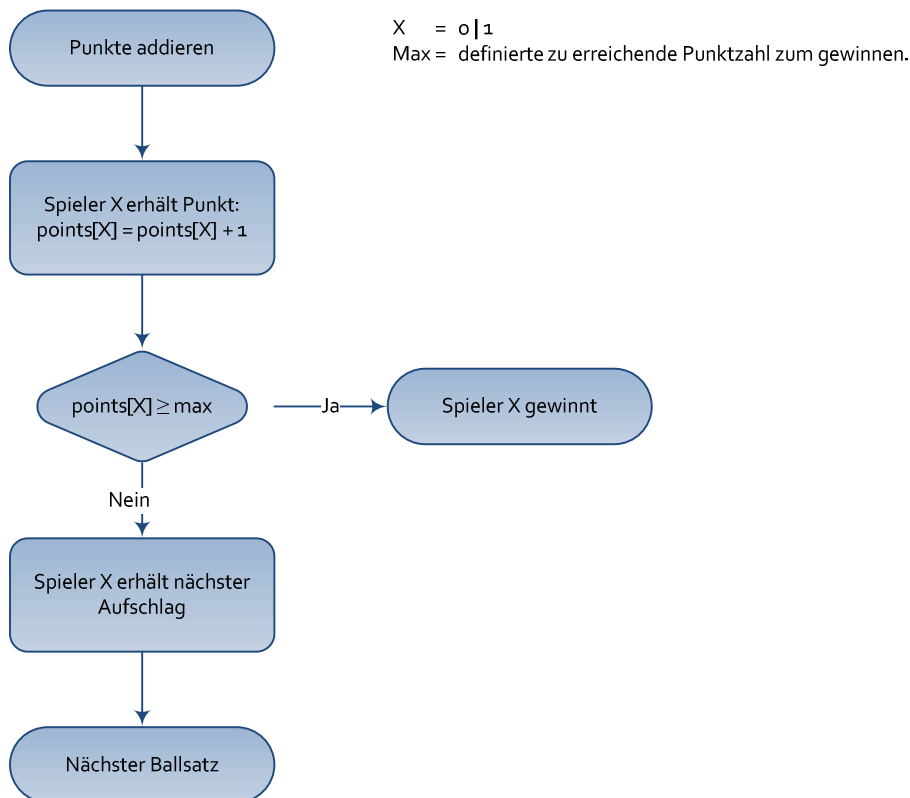


Abbildung 12: UseCase – Punktegewinn



7 Testing

Das Testing wird hauptsächlich über manuelle Tests mit Hilfe des haptischen Gerätes durchgeführt. Um den Spielablauf testen zu können, werden die Use-Cases (6.12 Use-Cases) als Vorlage verwendet. Die UseCases liefern dabei die Vorgänge und die zu testenden Resultate.

Einzelne Klassen, welche zum Beispiel den Use-Case Punktegewinn (siehe 6.12.4 Punkte addieren) betreffen, werden wir automatisiert testen lassen.

8 Abgrenzung

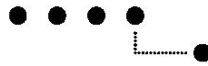
CAVE-System

Die Inbetriebnahme eines weiteren Haptic-Computers (Requirement 6.3.6) ist nicht Teil der BachelorArbeit. Sollte die Inbetriebnahme nach unserer Ansicht zuviel Zeit in Anspruch nehmen, behalten wir uns das Recht vor, die Priorität für im Zusammenhang-stehende Requirements herabzustufen.

I3D-Framework

Die Benutzung des I3D-Frameworks mit zwei oder n Haptic-Computer ist nicht Teil unserer Arbeit. Sollten wir nach unserer Ansicht zu viel Zeit mit der Integration verlieren, behalten wir uns das Recht vor, die Priorität für im Zusammenhang-stehende Requirements herabzustufen.

Die Integration der Haptic wurde im Rahmen einer anderen Bachelor-Arbeit gemacht. Die Integration weist Fehler auf. Im Rahmen unserer BA werden wir mit diesen Fehler konfrontiert sein. Sollten die Fehler zu schwerwiegend sein, werden ausschliesslich MotionTracking verwenden.



9 Projektmanagement

9.1 Einleitung

Für unsere Bachelor-Thesis wollen wir ein vollumfängliches 3D-Spiel umgesetzt haben, welches Mithilfe von haptischen Geräten oder Input eines Motion-Trackings gespielt werden kann. Um dies zu erreichen, haben wir uns bereits in dem Projekt 2 um die Einarbeitung in verschiedene Frameworks gekümmert. In diesem Projekt gilt es, das Gelernte umzusetzen. Um dieses hohe Ziel zu erreichen, haben wir uns an eine stricte Planung zu halten. Um dies zu vereinfachen, verwenden wir das Projektmanagementtool "Redmine"³, um die verschiedenen Meilensteine zu planen und Engpässe frühzeitig zu bemerken. Unser Tool läuft unter <http://denethor.bfh.ch/>. Die Detailbeschreibung der Aufgaben ist im Kapitel "6 Anforderungen" zu finden. Die Liste bezüglich der Prioritäten dieser Ziele ist anschliessend danach zu finden.

9.2 Organigramm

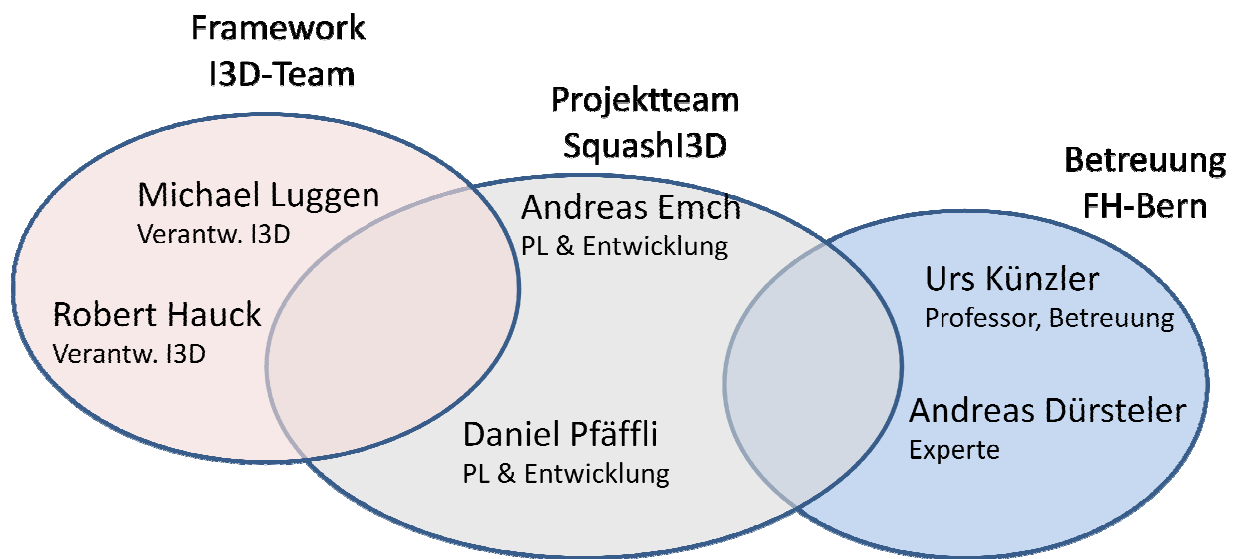


Abbildung 13: Organigramm

Rolle	Name	Kontakt
Betreuung	Urs Künzler	urs.kuenzler@bfh.ch
Experte	Andreas Dürsteler	
Projektleitung & Entwicklung	Andreas Emch	Emcha1@bfh.ch
Projektleitung & Entwicklung	Daniel Pfäffli	Pfafd1@bfh.ch



Rolle	Name	Kontakt
Unterstützung Framework	I3D- Michael Luggen	michael.luggen@bfh.ch
Unterstützung Framework	I3D- Robert Hauck	robert.hauck@bfh.ch

Tabelle 7: Projektrollen

Das Projektteam besteht aus zwei Personen. Aus diesem Grund gibt es keinen eindeutigen Projektleiter.

9.3 Ressourcen

Als zwingende Ressource ist der BFH-CAVE zu sehen. Die Entwicklung des Spiels wird zum grössten Teil auf den Geräten basieren, welche im CAVE auch zum Einsatz kommen. Abhängig von der Variante betrifft dies die MotionTracking-Geräte und/ oder die Haptic-Geräte.

Weiter wichtige Ressourcen:

Projektteam rund um das Framework I3D: Für Rückfragen werden Personen aus dem Projektteam benötigt, welche für das Framework zuständig sind.

9.4 Projektvorgehen

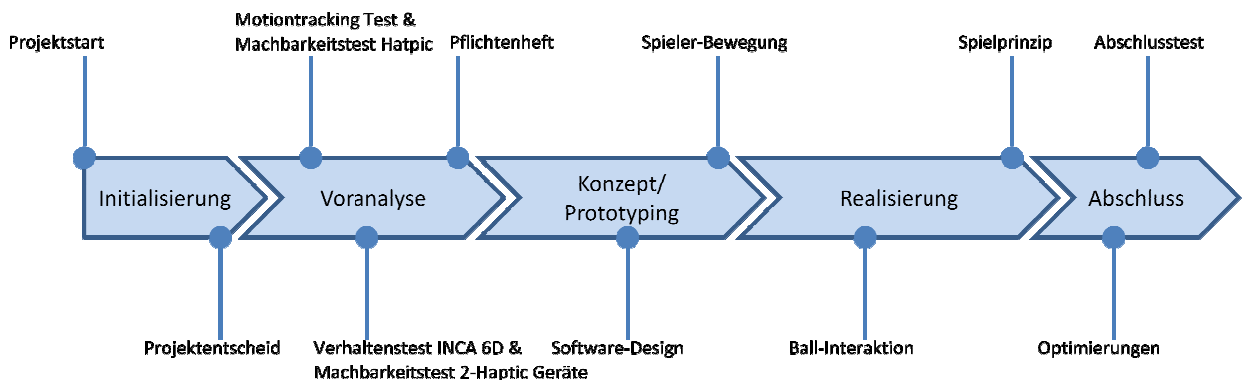


Abbildung 14: Projektphasen

PHASE	ZIEL	AUFGABEN
Initialisierung	Die erste Projektphase dient der Konstituierung des Projektteams (Organisation, Rollen, Grob-/Feinplanung), Festlegung der Projektprinzipien (Kommunikation, Reporting), sowie der Formulierung eines klar abgegrenzten Projektauftrages. Im Rahmen dieser Fachhochschul-internen Projektarbeit und durch die Projektgrösse werden auf viele Teile verzichtet.	<ul style="list-style-type: none"> •Projektabgrenzung / Projektumfang festlegen •Projektauftrag formulieren •Projektplanung durchführen <p>Die Arbeiten geschehen im Rahmen des Pflichtenheftes.</p>
Voranalyse	Nach erfolgter Projektinitialisierung werden die Anforderungen an die Lösung konkretisiert. Dabei sollen die Entscheidungsgrundlagen für	<ul style="list-style-type: none"> •Prozesse identifizieren •Funktionale Anforderungen formulieren



	die Umsetzung geschaffen werden.	<ul style="list-style-type: none"> •Schnittstellen identifizieren •Validierung der funktionalen Anforderungen <p>Die Aufgaben werden im Pflichtenheft festgehalten.</p>
Konzept/ Prototyping	Innerhalb des in der Voranalysephase definierten Rahmens werden die Detailspezifikationen (SRS) als verbindliche Vorgabe für die folgende Realisierungsphase erarbeitet und verabschiedet. Im Sinne eines Prototypings werden Zwischenversionen einzelner Lösungselemente als Machbarkeitsstudie getestet.	<ul style="list-style-type: none"> •Aufnahme detaillierter Anforderungen •Erstellung erster Prototypen •Erstellung und Pflege der Detailspezifikation (SRS) basierend auf den Resultaten der Prototypen <p>Die SRS wird im Rahmen einer technischen Dokumentation abgegeben.</p>
Realisierung	Die Detailkonzepte werden gemäss SRS (Software Requirement Specification) realisiert. Als Resultat der Phase steht eine lauffähige, getestete Lösung zur Verfügung.	<ul style="list-style-type: none"> •Beta Version der Lösung •Technische Dokumentation
Abschluss	Der Abschluss bildet die Abgabe des Projektes, sowie einer Präsentation der Lösung.	<ul style="list-style-type: none"> •Software und Dokumente •Präsentation

Tabelle 8: Projektphasen



9.5 Übersicht der Meilensteine

Für die Entwicklung verwenden wir das Prototyping-Verfahren. Die Prototypen sind so aufgebaut, dass sie jeweils auf den Vorgänger-Prototypen aufbauen. Diese Test-Schritte definieren zugleich unsere Meilensteine. Sie dienen der Planungssicherheit.

Die Prototypen werden folgenden Projektphasen zugewiesen.

Initialisierungsphase		
	Projektstart	17.09.2012
	Projektentscheid	21.09.2012
Voranalysephase		
9.5.5	MotionTracking Test	28.09.2012
9.5.2	Machbarkeitstest Haptic	28.09.2012
9.5.4	Verhaltenstest INCA 6D	05.10.2012
9.5.3	Machbarkeitstest 2-Haptic Geräte	05.10.2012
9.5.1	Pflichtenheft	12.10.2012
Konzept-/ Prototyping-Phase (ggf. auch oder separate Testphase, je nach Projekt)		
9.5.6	Software Design	26.10.2012
9.5.7	Spieler-Bewegungen	16.11.2012
Realisierungsphase		
9.5.8	Ball-Interaktion	30.11.2012
9.5.9	Spielprinzip	14.12.2012
Abschlussphase		
9.5.10	Optimierungen	28.12.2012
	Abschlusstest	04.01.2013
9.5.11	Endprodukt	18.01.2013

Tabelle 9: Meilensteine

9.5.1 Pflichtenheft

Der Meilenstein sieht die Abgabe des definitiven Pflichtenheftes vor.

9.5.2 Machbarkeitstest Haptic

Dieser Meilenstein dient der Abklärung der Implementation von Haptic in I3D. Für diesen Test wird das existierende Spiel Haptic-Pong im CAVE laufen gelassen. Die erwarteten Erkenntnisse fließen in den Varianten-Entscheid 6.6 mit ein.



9.5.3 Machbarkeitstest 2-Haptic Geräte

Dieser Test sieht die Integration eines weiteren Haptic-Computer in das CAVE-System vor. Für diesen Test wird die Hilfe von CAVE-Experten, wie Robert Hauck, Urs Künzler oder Michael Luggen, vorausgesetzt.

Für diesen Test wird ein Programm geschrieben, welches die Anzahl erkannter Haptic-Geräte anzeigt.

9.5.4 Verhaltenstest INCA 6D

Dieser Test sieht vor, die Freiheitsgrade des INCA 6D im CAVE kennenzulernen. Dafür wird eine Beispielapplikation, welche sich im I3D Framework befindet, verwendet.

Die gewonnen Erkenntnisse werden für die Planung des Squash-I3D verwendet.

Erwartete Erkenntnisse:

- Abschätzung der möglichen physikalischen Kräfte
- Physische Freiheiten eines Spielers mit dem INCA 6D

9.5.5 MotionTracking Test

Der Test dient der Analyse. Mit Hilfe einer Beispielapplikation, im I3D Framework vorhanden, werden die Möglichkeiten und Einschränkungen des MotionTrackings analysiert.

Erwartete Erkenntnisse:

- Einschränkungen in Bewegungen
- Geschwindigkeit der Erkennung einer Bewegung
- Bestehende Modelle, welche auf eine Squash-Figur angewandt werden können

9.5.6 Software Design

Der Meilenstein Software Design besteht aus einem ersten Teil der Dokumentation. Er beinhaltet das Software-Design:

- Klassendiagramm
- Definierte Public-Methods
- Notwendige Schnittstellendefinitionen
- Spezifikationsergänzungen zum Pflichtenheft

9.5.7 Spieler-Bewegungen

Der Prototyp dient uns dazu, das Spieler-Modell richtig auf das Spiel vorzubereiten. In den Prototypen laden wir den Charakter und werden diesen bereits im Prototypen 1 steuern können.

Folgende Features werden implementiert:

- Verwendung des I3D (6.3.1 Framework I3D)
- 6.4.1 Design Basic
- Modellierung des Spielers und eine programmierte Steuerung Animationen (6.4.2 Modellierung)
- Varianten-abhängige Steuerung: (6.3.5 Input - Motion Tracking, 6.3.4 Input - Schnittstelle Haptik, 6.5.3 Input - Geräte-Eigenschaften, 6.5.10 Input - Haptisches Feedback, 6.5.11 Input - Interaktion Haptic, 6.5.12 Input - Interaktion MotionTracking)



9.5.8 Ball-Interaktion

Der Prototyp dient der Interaktion zwischen Haptic, MotionTracking und der Physik-Engine. Folgende Features sind Bestandteil:

- Die Features von 9.5.7 Spieler-Bewegungen
- Kollisionserkennung: (6.5.7 Collisiondetection)
- Je nach Variante, Haptic: (6.5.10Input - Haptisches Feedback)

9.5.9 Spielprinzip

Dieser Prototyp baut auf den Vorhergehenden auf und erweitert diesen durch Hinzufügen des Balls. Auch werden die Regeln hier implementiert um den entsprechenden Spielstand anzuzeigen und zu halten.

Folgende Features werden implementiert:

- Verwendung des I3D (6.3.1 Framework I3D)
- Regeln sind implementiert um einen korrekten Spielstand zu gewährleisten. (4.1 Regeln)
- Audio wird abgespielt wie vorgegeben. (6.5.13 Audio)

Ebenfalls integrieren wir die Menüführung. Folgende Features werden implementiert:

- Das Menu wird implementiert und reagiert auf die Inputs vom Spieler. (6.5.14 Menüführung)
- Das Menu muss über dem aktuellen Spielstand gerendert werden. (6.4.3 Rendering)



9.5.10 Optimierungen

Dient als Schritt um das Squash-I3D auf den CAVE anzupassen. Die Lichtverhältnisse und Darstellungsoptimierungen wird dabei beachtet. Dieser Meilenstein dient dem Entscheid des Einsatzes der Shaders.

Features:

- Shaders: (6.4.4 Rendering Advanced)

9.5.11 Endprodukt

Umfasst mindestens eine minimale spielbare Version von SquashI3D.

9.6 Projektplanung

Für die gesamte Projektplanung haben wir ein Gant-Diagramm erstellt, um die Abhängigkeiten besser zu visualisieren.

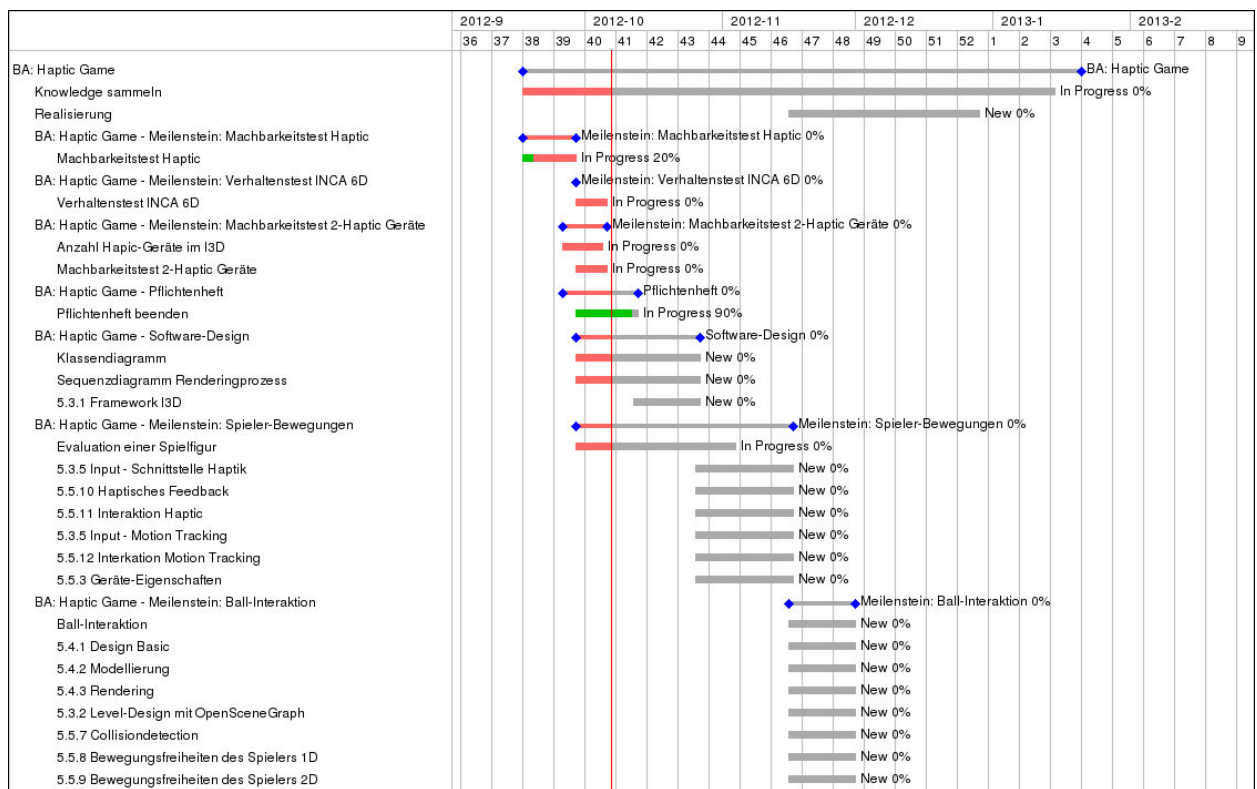


Abbildung 15: Gant-Diagramm –Teil 1

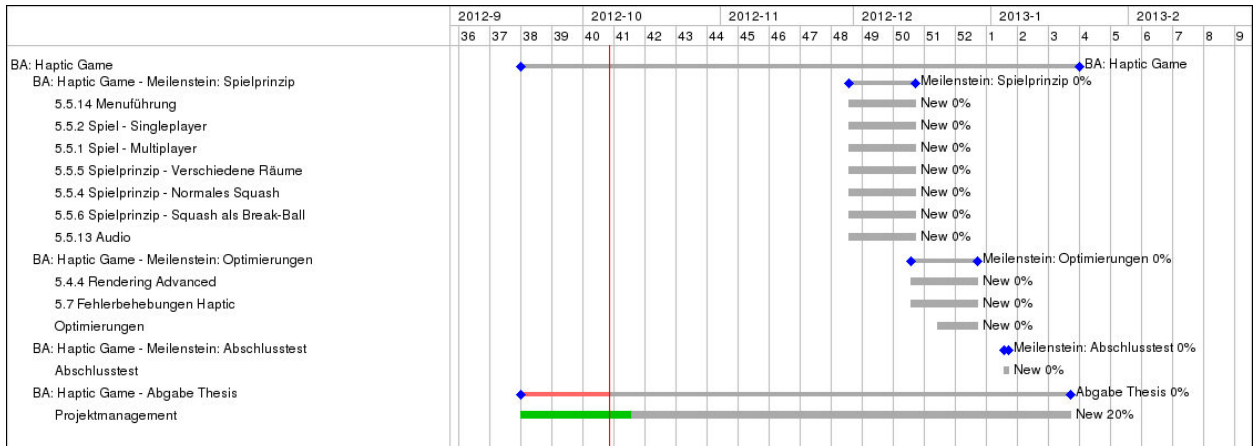


Abbildung 16: Gant-Diagramm –Teil 2

9.7 Projektorganisation

Die Projektsources und Versionierung verwalten wir im SVN. Die Struktur sieht folgendermassen aus:

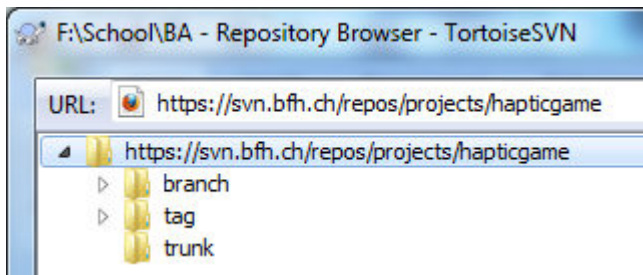


Abbildung 17: Aufbau SVN

branch	Verwenden wir, um an der bestehenden Lösung (trunk) keine riskanten Änderungen machen zu müssen.
tag	Beschreibt eine lauffähige Version.
trunk	Hauptarbeitsbereich, da wir nur zu Zweit daran arbeiten, werden wir direkt auf dem trunk Änderungen committen. Regeln: - Jeder Commit ist build-bar - Der Kommentar beinhaltet den Task woran gearbeitet wurde.

Tabelle 10: SVN

Dokumentation und Projektplanung geschieht im Redmine. Alle Planänderungen sind hier verfolgbar.

Link: <https://denethor.bfh.ch>



Issues - BA: Haptic Game - x
https://denethor.bfh.ch/projects/ba--haptic-game/issues

Home My page Projects Knowledge base Spent time Help Logged in as pfafd1 My

Bern University of Applied Sciences
Engineering and Information Technology
Institute for Human Centered Engineering - Project Management

Search: BA: Haptic G

OVERVIEW ACTIVITY ROADMAP **ISSUES** NEW ISSUE GANTT DOCUMENTS SETTINGS

BA: Haptic Game

Filters
Status: all Add filter

Options

Apply Clear Save

#	Tracker	Status	Subject	Author	Assignee	% Done
261	Support	New	Projektmanagement	Daniel Pfäffli [pfafd1]		
257	Task	New	Realisierung	Daniel Pfäffli [pfafd1]		
256	Task	New	Abschlusstest	Daniel Pfäffli [pfafd1]		
255	Task	New	Optimierungen	Daniel Pfäffli [pfafd1]		
254	Task	New	Spielprinzip	Daniel Pfäffli [pfafd1]		
253	Task	New	Ball-Interaktion	Daniel Pfäffli [pfafd1]		
252	Task	In Progress	Knowledge sammeln	Daniel Pfäffli [pfafd1]		
251	Task	In Progress	Verhaltenstest INCA 6D	Daniel Pfäffli [pfafd1]		
250	Task	In Progress	Machbarkeitstest 2-Haptic Geräte	Daniel Pfäffli [pfafd1]		
249	Task	In Progress	Machbarkeitstest Haptic	Daniel Pfäffli [pfafd1]		
248	Task	New	MotionTracking/ Haptic auf Figur anwenden	Daniel Pfäffli [pfafd1]		
247	Task	In Progress	Evaluation einer Spielfigur	Daniel Pfäffli [pfafd1]		

Issues
View all issues
Summary

Graphs
Open aging issues
Total issues over time
Gantt

Milestones

Abbildung 18: Redmine

¹ <http://www.redmine.org/>



10 Anhang

10.1 Historie der Dokumentversionen

Version	Datum	Autor	Änderungsgrund / Bemerkungen
0.1	21.09.2012	Andreas Emch, Daniel Pfäffli	Ersterstellung
0.2	26.09.2012	Andreas Emch, Daniel Pfäffli	Bemerkungen Urs Künzler, Überarbeitung Dokument.
0.3	28.09.2012	Daniel Pfäffli	Termine
0.9	07.10.2012	Andreas Emch, Daniel Pfäffli	Einfließen lassen der Bemerkungen von Herrn Künzler.
1.0	11.10.2012	Andreas Emch, Daniel Pfäffli	Keine Rückmeldung von Andreas Dürsteler/ Urs Künzler Vollständige Version
1.1	21.10.2012	Andreas Emch, Daniel Pfäffli	Rückmeldung von Herrn Künzler, 18.10.2012, in End-Version einfließen lassen.

Tabella 11: Versionierung