HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

INFORMATIK

INSTITUT
FÜR
SOFTWARE

Vier Gewinnt

Semesterarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2010

Autor(en):	Patrick Dünser / Amon Grünbaum
Betreuer:	Prof. Dr. Markus Stolze
Projektpartner:	-
Experte:	-
Gegenleser:	Prof. Eduard Glatz

Teil I
Allgemeine Informationen

1 Abstract

Der Info-Tag ist der Werbe-Event für zukünftige Studenten an der HSR. Für diesen Event soll ein neues interaktives und attraktives Ausstellungsobjekt gemeinsam von Studenten der Abteilungen M, E, und I entwickelt werden. Das Ausstellungobjekt ist ein Roboter gegen den der Infotag-Besucher Vier Gewinnt spielen kann. Für diesen Zweck wurden wir beauftragt ein User Interface zu entwerfen, den Algorithmus für die künstliche Intelligenz und eine Schnittstelle zum Roboter zu implementieren.

Ziel war es, die Webseite übersichtlich zu gestaltet und somit die Bedienung sehr simpel und intuitiv zu machen. So dass auch nicht informatikaffine Leute ohne Probleme und ohne Erklärung sich gleich zu Recht finden und spielen können. Wir haben die Webseite mit HTML5 / CSS3 gemacht. So kann jeder Infotag-Besucher mit seinem eigenen internetfähigen Mobiltelefon Vier Gewinnt gegen den Computer oder einen anderen Infotag-Besucher spielen. Alle Besucher, die kein internetfähiges Mobiltelefon besitzen, können auf einem iPad am Roboter spielen. HTML5 bietet viele neue Möglichkeiten um Inhalte im Web leichter darzustellen. Wir nutzten die Möglichkeit mit AJAX ein Server-Push zu implementieren, damit der Web-Server geänderte Inhalte der Webseiten dem Benutzer sofort mitteilt. Die künstliche Intelligenz des Computers wurde mit dem Minimax-Algorithmus mit Alpha-Beta-Pruning realisiert. Die Webseite läuft auf einem eigenen Server. Dieser ist an einem Access-Point angeschlossen, welcher das WLAN „HSR-VierGewinnt“ zu Verfügung stellt. Auf dem Server läuft Microsoft Windows 2008 Server mit integriertem DNS und einem Apache / Tomcat Server. Nun hoffen wir, dass der Roboter in naher Zukunft fertiggestellt wird, damit unsere Software auch verwendet wird.

Inhaltsverzeichnis

2 Management Summary

2.1 Ausgangslage

Der Info-Tag ist der Werbe-Event für zukünftige Studenten an der HSR. Für diesen Event soll ein neues interaktives und attraktives Ausstellungsobjekt gemeinsam von Studenten der Abteilungen M, E, und I entwickelt werden. Das Ausstellungobjekt ist ein Roboter gegen den der Infotag-Besucher Vier Gewinnt spielen kann. Aufbauend auf dem Input aus dem im FS 2010 durchgeführten Ideenwettbewerb wurde entschieden, ein Gerät zu entwickeln, welches das Spielen von Vier Gewinnt erlaubt. Unterschiedliche Spielmodi sollen möglich sein, unter anderem ein Modus in dem ein Benutzer gegen den Computer spielt. Dabei sollen unterschiedliche Stärken des Computers gewählt werden können. Der Informatik Teil der Aufgabe besteht in der Implementation des Algorithmus, der die Züge plant. Der Output des Systems (der nächste Zug) sollte als Signal auf einem USB Port ausgegeben werden. Der Algorithmus zur Spielplanung soll als MiniMax oder Branch-and-Bound Algorithmus implementiert werden.

Für diesen Zweck wurden wir beauftragt ein User Interface zu entwerfen, den Algorithmus für die künstliche Intelligenz und die Schnittstelle zum Roboter zu implementieren.

2.2 Probleme

Folgende Probleme haben sich uns gestellt: In der Informatik passieren sehr viele Vorgänge im Hintergrund, in den Tiefen der Hardware, welche normale Leute nie sehen. Uns stellten sich darum die Fragen: Wie bringt man den Leuten die aktuelle Technik nahe? Wie macht man sie greifbar?

Wir wollten den Infotag-Besuchern Informatik nahe bringen, auch solchen die kein grosses Interesse an Informatik haben. Und Informatik einmal von der etwas anderen Seite präsentieren.

2.3 Lösung

Wir haben es mit einer für iPhone & iPad optimierten HTML5/CSS3-Webseite gemacht. So kann jeder Infotag-Besucher mit seinem eigenen internetfähigen Mobiltelefon Vier Gewinnt gegen den Computer oder einen anderen Infotag-Besucher spielen. Die künstliche Intelligenz des Computers wurde mit dem Minimax-Algorithmus mit Alpha-Beta-Pruning realisiert. Alle Besucher, welche keine internetfähiges Mobiltelefon besitzen, können auf einem iPad am Vier Gewinnt-Roboter spielen.

Die Webseite ist übersichtlich gestaltet und somit ist die Bedienung sehr simpel und intuitiv. So dass auch nicht informatikaffine Leute ohne Probleme und ohne Erklärung gleich navigieren und spielen können. Wir haben uns intensiv den Web-Technologien HTML5, CSS3 und AJAX auseinander gesetzt. HTML5 bietet viele neue Möglichkeiten um Inhalte im Web leichter darzustellen, vor allem Multimedia-Inhalte. Wir nutzten die Möglichkeit von AJAX um ein

Server-Push zu implementieren, damit der Web-Server geänderte Inhalte der Webseiten dem Benutzer sofort mitteilt. CSS3 bietet für eine schöne, sauber Gestaltung der Webseite einige nützliche Möglichkeiten. Wir nutzten diese um damit neue Buttons (mit Schatten bzw. Farbverlauf) oder abgerundete divs zu gestalten.

Die Webseite läuft auf einem eigenen Server und ist an einem Access-Point angeschlossen, welcher das WLAN „HSR-VierGewinnt“ zu Verfügung stellt, damit dem Benutzer keine Kosten für den Datentransfer anfallen. Damit der Access-Point optimal funktioniert, mussten wir diesen zuerst flashen und ein neues Betriebsystem installieren. Nun funktioniert das Portforwarding einwandfrei. Auf dem Server läuft Microsoft Windows 2008 Server mit integriertem DNS und einem Apache / Tomcat Server. Ein eigener DNS war nötig, damit man auch ohne Internetanschluss auf die Webseite viergewinnt.hsr.ch zugreifen kann. Der Infomations-Dienst muss nur den Access-Point und den Server einschalten und schon kann man auf die Homepage viergewinnt.hsr.ch zugreifen.

2.4 Lessons Learned

Für unser nächstes Projekt werden wir früher Code-Reviews durchführen, damit wir kleinere Codeausschnitte und somit auch weniger Refactoring aufs Mal machen müssen. Des Weiteren werden wir uns umschauen, ob es für ein zweier Team noch bessere Vorgehensmodelle als Scrum gibt. Beim nächsten Projekt werden wir sicher wieder LaTeX für die Dokumentation verwenden, auch wenn wir lange gebraucht haben, um damit klar zu kommen. Dafür entfällt danach das lästige Formatieren.

2.5 Erfahrungsbericht Amon Grünbaum

Gereizt hat mich an diesem Projekt das Zusammenspiel der verschiedenen Technologien und das interdisziplinäre Arbeiten mit verschiedenen Studienrichtungen.

Die Semesterarbeit wurde relativ kurzfristig aufgesetzt und zu Beginn gab es noch keine Teams der anderen Abteilungen, mit welchen wir das Projekt starten konnten. Trozdem begannen wir mit den Vorarbeiten zum Projekt. Kurz nach dem Start wechselte unser Gemüt jedoch, als bekannt wurde, dass Renato Müller, welcher Maschinentechnik studiert, auch am Projekt mitarbeitet und den Roboter realisiert werden würde. Dies motivierte uns sehr. Dass kein Elektrotechniker gefunden wurde, fanden wir zwar schade, dachten jedoch, dass wir das Projekt auch ohne ihn zu Ende bringen können. Die interdisziplinäre Arbeit funktionierte ziemlich gut und wir trafen uns mehrmals mit allen Beteiligten für konstruktive Meetings.

Die Teamarbeit zwischen Patrick Dünser und mir funktionierte sehr gut. Wir arbeiteten praktisch immer in der Schule und konnten Probleme gleich gemeinsam besprechen und dadurch schnell beheben. Meinungsverschiedenheiten gab es eigentlich nie und durch die unterschiedlichen Stärken konnten wir beide von einander profitieren.

Die Struktur mit wöchentlichen Reviews mit Prof. Dr. Markus Stolze fand ich ebenfalls sehr gut. So konnten wir offene Fragen gerade besprechen und die erledigte Arbeit reflektieren.

Natürlich lief im Projekt nicht immer alles perfekt. Manche technische Herausvorderung brauchte länger als ursprünglich geplant und so kamen wir auch ab und zu ein wenig in Verzug. Teilweise gerieten wir mit den Sprint Reviews von Scrum ein wenig in Rückstand, was für das gesamte Projektmanagement nicht optimal war. Überhaupt denke ich hat das Arbeiten nach Scrum nicht immer ganz richtig funktioniert. In einer nächsten Arbeit würde ich dies von Anfang an besser aufsetzen, um für das gesamte Projekt einen besseren Überblick zu bewaren.

In der zweitletzten Woche hatten wir einen Review des Codes und des User Interfaces mit Silvan Gehrig und Kevin Gaunt. Die beiden gaben uns sehr gute Hinweise, was wir noch verbessern sollten. In einem nächsten Projekt würde ich es begrüssen, früher solch ausfürliche Code- und UI-Reviews mit Dozenten oder Assistenten durchzuführen.

Mit unserm Resultat bin ich zufrieden. Natürlich kann mann immer noch etwas mehr Implementieren und ein paar optionale Features mehr wären super gewesen. Da aber vieles mehr Zeit in Anspruch genommen hat, war dies leider nicht mehr möglich. Wir konnten jedoch alle zwingenden Komponenten implementieren. Diese Komponenten haben wir nicht einfach irgendwie implementiert, sondern sauber programmiert und gut getestet.

Ein kleiner dämpfer war, als wir erfuhren, dass der Roboter nicht plangemäss fertiggestellt werden kann und wir unsere Arbeit nicht vollständig abgeben können. Dies hat sicher mit der Tatsatche zu tun, dass für den Elektrotechnikteil keine Studenten gefunden wurden. Aber auch im Maschinentechnikteil gab es zusätzliche Verzögerungen, sodass der Roboter bis zum Semesterende nicht gebaut werden konnte. Ich hoffe sehr, dass der Roboter doch noch gebaut wird, damit sich unser Einsatz gelohnt hat.

Die Semesterabrbeit war eine anstrengende, aber sehr interessante Arbeit und eine gute Erfahrung. Ich möchte mich bei allen Beteiligten für die gute Zusammenarbeit bedanken.

2.6 Erfahrungsbericht Patrick Dünser

Die Semesterarbeit war sehr lehrreich und machte Spass, auch wenn der Roboter leider nicht in diesem Semster gebaut wird. Später werde ich darauf eingehen. In dieser Semesterarbeit konnte ich einerseits viele neue Technologien kennen lernen und andererseit mein bereits vorhandes Wissen über diverse Technologien und Server vertiefen. Zu den, für mich neuen Technologien zählen HTML5, CSS3 und AJAX. HTML5 bietet viele neue Möglichkeiten um Inhalte im Web leichter darzustellen, vorallem Multimedia Inhalte. Bei CSS3 ist die Platzierung der Elemente immernoch aufwendig, wie bei seinem Vorgänger, bietet allerdings designmässig sehr viele Verbesserungen, die wir erfolgreich angewandt haben.

Die Teamarbeit mit Amon Grünbaum war sehr gut. Wir verstanden uns während des ganzen Projektes super, die Arbeitsaufteilung klappte einwand-

frei und wir ergänzen uns sehr gut. So das wir auch unsere Bachelor-Arbeit zusammen in Angriff nehmen werden.

Nun will ich noch kurz die negativen Punkte in diesem Projekt ansprechen. Schon zu Beginn des Projekts war nicht sicher, ob Studenten von den Studiengängen M und E für das Projekt gefunden werden können. Der Studiengang M konnte mit Renato Müller einen kompetenten Student aufwarten. Leider gab es keinen E-Studenten, welcher sich für das Projekt begeistern konnte. So mussten wir zusammen mit Renato Müller versuchen die Lücke zwischen M und I zu schliessen. Was wir allerdings mit etwas Mehraufwand geschafft haben.

Leider konnte Renato Müller seine Pläne nicht in Auftrag geben, da er von der Schule keine finanziellen Mittel zugesprochen bekam. Was meiner Meinung hauptsächlich an der nicht ganz optimalen Kommunikation zwischen Autraggebern und Schule lag. Somit haben wir "nur" ein normales Vier Gewinnt programmiert, ohne das Resultat auf einem Roboter zu bestaunen.

Nichts destotrotz haben wir uns von diesen Rückschlägen nicht klein kriegen lassen und haben eine saubere Arbeit abgegeben, welche die Schnittstelle beinhaltet, damit, wenn der Roboter in Zukunft fertig gestellt werden sollte, dieser ohne weiteres an den Server angeschlossen werden kann und alles läuft.

Ich habe Vieles über das Vorgehensmodell Scrum gelernt. Allerdings werde ich mich fürs nächste Projekt umschauen, ob es für kleine Teams noch ein besseres Vorgehensmodell gibt. Während der Arbeit habe ich sehr viel über die beiden Algorithmen Branch-and-Bound und Minimax gelernt. Zudem bin ich erstaunt, wievel Möglichkeiten HTML5 und CSS3 im Gegensatz zu deren Vorgänger-Versionen bieten, um eine Webseite sauber zu gestalten.

3 Aufgabenstellung

3.1 Auftraggeber und Betreuer

Diese Studienarbeit hat keinen direkten Auftraggeber. Als interimistischer Auftraggeber fungiert das Institut für Software.

- Ansprechpartner Auftraggeber: (NA)
- Betreuer HSR: Prof. Dr. Markus Stolze, Institut für Software mstolze@hsr.ch

3.2 Ausgangslage & Zielsetzung

Der Info-Tag ist der Werbe-Event für zukünftige Studenten an der HSR. Für diesen Event soll ein neues interaktives und attraktives Ausstellungsobjekt gemeinsam von Studenten der Abteilungen M, E, und I entwickelt werden.

Aufbauen auf Input aus dem im FS 2010 durchgeführten Ideenwettbewerb wurde entschieden ein Gerät zu entwickeln welches das Spielen von „4-Gewinnt“ erlaubt (http://de.wikipedia.org/wiki/Vier_gewinnt)

Unterschiedliche Spielmodi sollen möglich sein, unter anderem ein Modus in dem ein Benutzer gegen den Computer spielt. Dabei sollen unterschiedliche Stärken des Computers gewählt werden können.

Der Informatik Teil der Aufgabe besteht in der Implementation des Algorithmus der die Züge plant. Der Output des Systems (der nächste Zug) sollte als Signal auf einem USB Port ausgegeben werden. Der Algorithmus zur Spielplanung kann relativ einfach als MiniMax, Branch-and-Bound Algorithmus implementiert werden.

Damit die Aufgabe in der Komplexität einer Studienarbeit angemessen ist, sind die Studenten gehalten weitere Ausbauschritte zu planen. Im Folgenden eine Liste von Ideen:

- Möglichkeit der Zug-Eingabe und Überprüfung des Spielstandes über das Web oder ein Mobilgerät (Android, Win 7 Phone, iPhone)
- Visuelle Illustration des Suchalgorithmus
- Leader-Board z.B. mit Facebook Integration
- Zufallselemente: Ein vom Benutzer (oder auch System) eingegebener Zug wird nicht immer korrekt ausgeführt. Z.B. in 25% der Fälle wird der Chip in ein Fach weiter links oder rechts fallen gelassen. Diese Zufallselement macht es schwieriger das Spiel zu planen.

Es bleibt auch zu entscheiden ob der aktuelle Stand des Spiels jeweils eingelesen wird (z.B. durch Erkennung der Steine auf einem Bild) oder ob angenommen wird, dass das System selber korrekt über die Situation Buch halten kann.

Wahrscheinlich sind Anpassungen und Detailplanungen notwendig, so dass das Informatik-System sich in die mit E und M Studenten zusammen zu erarbeitende Gesamtlösung möglichst gut einpasst.

3.3 Zur Durchführung

Mit dem HSR-Betreuer finden in der Regel wöchentliche Besprechungen statt. Zusätzliche Besprechungen sind nach Bedarf durch die Studierenden zu veranlassen.

Alle Besprechungen sind von den Studenten mit einer Traktandenliste vorzubereiten und die Ergebnisse in einem Protokoll zu dokumentieren, das dem Betreuer E-Mail zugestellt wird. Zudem wird eine Projektseite ausgehend von Prof. Stolzes internen SA 2010 Wiki gemacht:

http://sinv0002.hsr.ch/MarkusStolze/wiki.cgi?StolzeArbeitenHS2010

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsresultate in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsresultate erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt aufgrund der am Abgabetermin abgelieferten Dokumentation. Die Evaluation erfolgt aufgrund des separat abgegebenen Kriterienkatalogs in Übereinstimmung mit den Kriterien zur SA Beurteilung. Es sollten hierbei auch die Hinweise aus dem abgegebenen Dokument „Tipps für die Strukturierung und Planung von Studien-, Diplom- und Bachelorarbeiten“ beachtet werden.

3.4 Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen. Die zu erstellenden Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Die Dokumentation ist vollständig auf CD/DVD in 2 Exemplaren abzugeben. Zudem ist eine kurze Projektresultatdokumentation im Wiki von Prof. Stolze zu erstellen dies muss einen Link auf einen YouTube Video enthalten welche das Resultat der Arbeit dokumentiert.

3.5 Termine

Siehe auch Terminplan auf

https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html

20.09.10 Beginn der Studienarbeit, Ausgabe der Aufgabenstellung durch die Betreuer.

20.12.10 Die Studierenden senden folgende Dokumente der Arbeit per Email zur Prüfung an ihre Betreuer: - Abstract/Kurzfassung - A0-Poster Vorlagen stehen unter den allgemeinen Infos Diplom-, Bachelor- und Studienarbeiten zur Verfügung.

23.12.10 Die Studierenden senden das vom Betreuer abgenommene und freigegebene Abstract/Kurzfassung als Word-Dokument an das Studiengangsekretariat (cfurrer(at)hsr.ch).

23.12.10 Abgabe des Berichtes an den Betreuer bis 17.00 Uhr.

3.6 Beurteilung

Eine erfolgreiche SA zählt 8 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von ca. 25 bis 30 Stunden budgetiert. Entsprechend sollten ca. 240h Arbeit für die Studienarbeit aufgewendet werden. Dies entspricht ungefähr 17h pro Woche (auf 14 Wochen) und damit ca. 2 Tage Arbeit pro Woche.

Für die Beurteilung sind der HSR-Betreuer verantwortlich unter Einbezug des Feedbacks des Auftraggebers (welches in diesem Fall entfällt)

Die Bewertung der Arbeit erfolgt entsprechend der verteilten Kriterienliste

Rapperswil, den 20. September 2010 Prof. Dr. Markus Stolze Institut für Software Hochschule für Technik Rapperswil

Teil II
Projekt Management

4 Projektorganisation

4.1 Autoren

Diese Semesterarbeit ist lediglich ein Teil des Projektes Vier Gewinnt. Folgend die beiden Autoren dieser Semesterarbeit:

(a) Amon Grünbaum

(b) Patrick Dünser

Abbildung 1: Autoren

4.2 Projektteam

Das komplette Pojekt Vier Gewinnt umfasst eine Semesterarbeit der Studienrichtung Informatik, sowie eine Semesterarbeit der Studienrichtung Maschinentechnik.

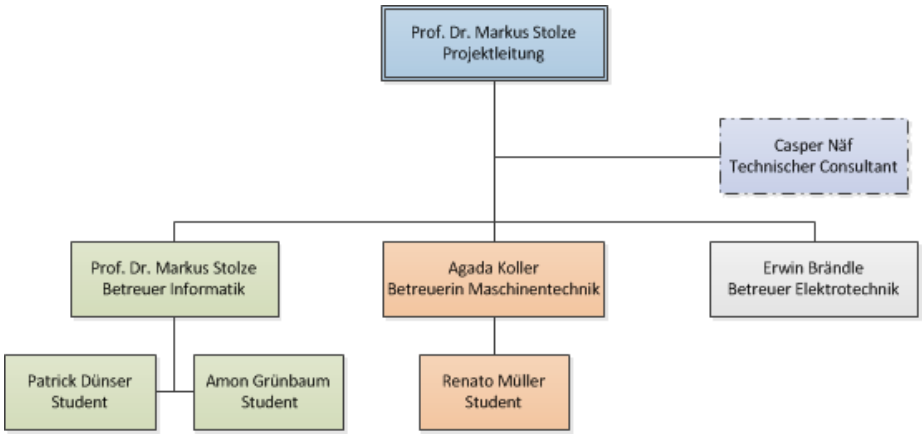


Abbildung 2: Projektteam

Prof. Dr. Markus Stolze übernimmt die Gesamtprojektleitung und ist zudem Betreuer für den Informatik-Teil.

Agathe Koller ist die Betreuerin für den Maschinentechnik-Teil.

Erwin Brändle hätte die Betreuung für den Elektrotechnik-Teil übernommen. Leider fand sich kein Student, welcher diese Arbeit als Semsterarbeit erledigen wollte.

Casper Näf ist der Technische Consultant. Er stellt sein Wissen im Bereich der Elektrotechnik zur Verfügung und half bei der Koordination zwischen Maschinentechnik und Informatik.

5 Vorgehensmodell

Folgendes Kapitel liefert eine kurze Übersicht über die Vorgehensmodelle Scrum. Die Kapitel über Scrum sind eine Kurzzusammenfassung der detaillierten Ausführungen der Web-Seite Scrum-Master.de.

5.1 Scrum

Scrum (engl. das Gedränge) ist ein Vorgehensmodell, welches beim Entwickeln von Software im Rahmen agiler Softwareentwicklung hilfreich ist. Die Teammitglieder organisieren sich weitgehend selbst und bestimmen auch die eingesetzten Software- Entwicklungswerkzeuge und -Methoden. Grundsätzlich verfolgt Scrum das Ziel, möglichst schnell eine ausführbare Software zu erhalten. Scrum enthält Empfehlungen für Rollen, Tätigkeiten (Meetings) und Artefakte, welche in den folgenden Unterkapiteln kurz beschrieben werden.

5.1.1 Scrum Rollen

Rolle	Beschreibung
Product Owner	• Definiert Anforderungen als "User Stories" • Vergibt Prioritäten (-> wählt Features aus)
Scrum Master	• Leitet Entwicklung und Meetings • Kontrolliert Qualität und Fortschritt
Scrum Team	• Optimalerweise 5-7 Personen • Interdisziplinär • Schätzt Entwicklungsaufwand • Organisiert sich selbständig

5.1.2 Scrum Tätigkeiten

Tätigkeit	Beschreibung

Planning	Jeder Sprint beginnt mit dem Sprint Planning, bei dem das Arbeitspaket des Scrum-Teams für den kommenden Sprint (Sprint Backlog) geschnürt wird. Vorgehen: 1. Product Owner überarbeitet Product Backlog 2. Team schätzt Aufwand für User Stories 3. Team teilt User Stories in Tasks auf -> Sprint Backlog
Daily Meeting	Das kurz (15 Minuten) zu haltende Daily Meeting findet an jedem Arbeitstag im Stehen statt und dient dem Team dazu, sich abzustimmen und gegenseitig zu informieren. Der Scrum Master nimmt teil, notiert sich genannte Impediments und greift nur moderierend ein, wenn es unbedingt nötig ist. Der Scrum Owner kann nach Möglichkeit teilnehmen, um auf dem neusten Stand zu bleiben und bei Bedarf Fragen zu beantworten. Das Scrum Team berichtet sich gegenseitig über folgendes: • Was habe ich seit dem letzten Daily Scrum getan? • Was plane ich, bis zum nächsten Daily Scrum zu tun? • Was hat mich bei der Arbeit behindert?

Sprint Review	Am Ende jedes Sprints präsentiert das Team dem Product Owner und allen interessierten Stakeholders das Ergebnis seiner Arbeit live am System und sammelt Feedback ein. Wichtige Punkte: • Das Team selbst präsentiert live am System, was es innerhalb des Sprints erreicht hat. • Keine Dummies, kein Power Point! Nur fertige Produktfunktionalität darf vorgeführt werden. • Auf Basis des Gezeigten entscheidet später der Product Owner, ob das Inkrement produktiv gesetzt oder weiter entwickelt werden soll. Diese Möglichkeit hat er nach jedem Sprint.
Sprint Retrospective	Das Sprint Retrospective folgt im Anschluss an das Sprint Review. Das Team diskutiert rückblickend auf den soeben zu Ende gegangenen Sprint und überlegt sich, was weshalb gut oder schlecht gelaufen ist und wie man den nächsten Sprint produktiver und/oder angenehmer gestalten kann.

5.1.3 Scrum Artefakte

Artefakt	Beschreibung
Product Backlog	Das Product Backlog enthält alle Features des zu entwickelnden Produkts in Form von User Storys. Vor jedem Sprint werden die einzelnen User Storys neu bewertet und priorisiert.
Sprint Backlog	Das Sprintbacklog enthält alle aus dem Product Backlog abgeleiteten Aufgaben, welche in einem Sprint bearbeitet werden. Eine Aufgabe sollte nicht länger als 16 Stunden dauern. Bei der Planung werden nur so viele Aufgaben geplant, wie einem Team an Kapazität zur Verfügung steht.
Sprint Burndown Chart	Der Burndown Chart stellt den Fortschritt des aktuellen Sprints dar. Die Kurve sollte am Ende des Sprints auf 0 sein. Die Tendenz der Kurve soll zeigen, ob der geschätzte Aufwand bis Ende Sprint erledigt werden kann.

Folgende Abbildung zeigt, wie die erwähnten Scrum Tätigkeiten und Artefakte zusammenhängen:

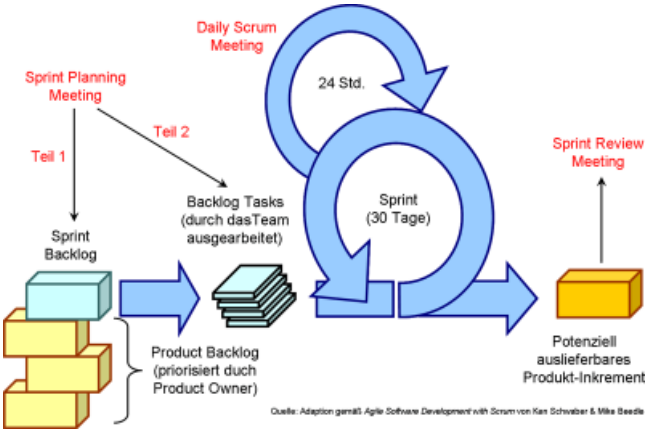


Abbildung 3: Scrum Ablauf

6 Visiondokument

6.1 Team

6.1.1 Mitglieder Informatik

Patrick Dünser	PD	pduens@hsr.ch
Amon Grünbaum	AG	agruenba@hsr.ch

6.1.2 Mitglieder anderer Studiengänge

Renato Müller	RM	r4muelle@hsr.ch

6.1.3 Betreuung und Bewertung

Prof. Dr. Markus Stolze	MS	markus.stolze@hsr.ch

6.1.4 Gegenleser

Prof. Eduard Glatz	EG	eglatz@hsr.ch

6.2 Aufgabenbeschreib

Der Inhalt dieser Semesterarbeit (SA) soll daraus bestehen, die im Kapitel 3 Aufgabenstellung vorgegebene Aufgabe zu erarbeiten. Im Rahmen dieses Projektes soll ein Vier Gewinnt Roboter für Informationstage an der Hochschule Rapperswil (HSR) erstellt und programmiert werden. Der Informatikteil dieser Aufgabe besteht darin eine Schnittstelle für die Robotersteuerung zu definieren, sowie die Intelligenz des Computers zu implementieren. Zusätzlich soll die interaktive Ansteuerung des Roboters über das eigene Mobilgerät der Informationstagbesucher einen Hauptteil der Arbeit einnehmen. Die Arbeit soll interdisziplinär mit Studenten anderer Studiengängen erarbeitet werden und besonders in der Ideenfindung gemeinsam verwirklicht werden. Am Schluss werden die verschiedenen Teile der Arbeit jedoch separat pro Studienrichtung bewertet.

6.3 Minimalanforderung

Der Mindestumfang des Projekts besteht aus einer funktionierenden Schnittstelle (USB) um einen Roboter anzusteuern und um den Zustand des Roboters auszulesen. Der Roboter muss in verschiedenen Spielstärken gegen einen Benutzer spielen können und in der schwierigsten Stufe eigentlich fast immer gegen einen Menschen gewinnen. Die Zugeingabe muss über ein touchfähiges Endgerät und über ein Touchgerät (iPad, iPod, etc) an der Vitrine möglich sein. Der Server sollte auch ohne Internetanschluss erreichbar sein.

6.4 Leistungsanforderungen

Neben den Minimalanforderungen gibt es auch noch andere Evaluationskriterien, welche für das Projekt eingehalten werden müssen. Folgende Kriterien sind besonders wichtig für das Projekt:

- autonom - Der Roboter muss ohne menschliche Hilfe mehrere Spiele nacheinander spielen können. (Zum Beispiel ohne das jemand immer wieder die Spielsteine aufräumt)
- transportabel - Der Roboter und der Server müssen einfach transportierbar sein und an einem anderen Ort eingesetzt werden können.
- minimale Wartung - Der Roboter soll ohne eine tägliche Wartung funktionieren.
- einfache Bedienung - Die Bedinung soll selbsterklährend sein.

6.5 Realisierung

6.5.1 Übersicht

Die Arbeit ist in verschiedene Teile unterteilt, in denen verschiedene Technologien eingesetzt werden. Das gesamte Projekt wird nach der SCRUM Methodik durchgeführt.

6.5.2 Schnittstelle zum Roboter und Spiellogik

Das Austellungsobjekt beinhaltet einen Server, auf dem der Webserver läuft und der Algorithmus für die Spiellogik.

- Webserver
  - Java
- Algorithmus / Spiellogik
  - MiniMax
  - Branch-and-Bound
- Webauftritt
  - HTML5
  - GUI optimierte für "Touchscreen-Eingaben"
  - Zugeingabe
    - mobilesEndgerät (IPhone / Android Phone)
    - Touchscreen am Objekt (optional)
    - PC per Browser
  - Multiplayerspiel von verschiedenen Endgeräten aus. (optional)

6.6 Partner / Auftraggeber

Die Arbeit wurde von den Instituten ILT (Institut für Labortechnologie), IFS (Institut für Software) sowie IMES (Institute for Microelectronics and Embedded Systems) ausgeschrieben. Diese drei Institute sind somit die Auftraggeber für das Projekt. Für den Informatikteil des Projekts ist das Institut für Software der Hauptauftraggeber.

Ebenfalls interessiert am Ausgang des Projektes werden die Informationsdienste der HSR sein. Sie führen jeweils die Infotage an der HSR durch, an welchen der Vier Gewinnt Roboter ausgestellt wird.

6.7 Konkurenz

Das Ziel dieser Arbeit ist es, das Studieren an der HSR anhand eines interessanten Ausstellungsobjekts für potentielle neue Studenten greifbar und schmackhaft zu machen. Da dieses Objekt die HSR representieren sollte, macht es für die Hochschule keinen Sinn, dieses Projekt extern realisieren zu lassen oder auf eine bestehende Lösung zurück zu greifen. Wenn es allerdings bereits eine Lösung von einer anderen Hochschule gibt, die die gleiche Leistungsanforderung hat, macht es keinen Sinn das Projekt durchzuführen. Es gibt bisher eine Lösung der Universität Stuttgart. Diese Lösung basiert auf Lego-Mindstorms. Das Resultat dieses Projekts ist sehr gut. Jedoch sind unsere Anforderungen unterschiedlich. Beim Projekt in Stuttgart war es wichtig, dass der Mensch selber seine Züge macht und der Spieler teilte dem Roboter jeweils mit, wann dieser den nächsten Spielzug ausführen muss. Die Anforderungen in unserem Projekt (beschrieben unter 6.4 Leistungsanforderungen) gehen jedoch viel mehr darum, dass der Roboter ohne grosse Wartung mehrere Spiele spielen kann. Die Aufgabenstellung ist somit nicht vergleichbar.

Weiter gibt es auch noch einen Vier Gewinnt Roboter der Hochschule Chur. Dieser unterscheidet sich ebenfalls stark von unserer Implementation. Zudem ist dieser Roboter auch schon älter und wird nicht mehr an Ausstellungen gezeigt.

Wir haben einen gefrästen, mit CAD konstruierten Roboter, der die Züge des Computers, sowie des menschlichen Gegners erledigt. Zudem werden die Züge des menschlichen Gegners per externer Quelle, das heisst über ein mobiles Endgeräte, eingegeben.

Eine Nichtdurchführung des Projektes wäre für die HSR möglich. Das Ziel eines attraktiven Ausstellungsobjektes, an dem Studenten aus verschiedenen Studienrichtungen interdisziplinär gearbeitet haben, könnte damit aber nicht erreicht werden.

6.8 Szenarios

Zur Veranschaulichung der verschiedenen Anforderungen haben wir drei Szenarien erstellt, in denen beispielhaft gezeigt wird, was erreicht werden sollte.

6.8.1 Minimum Szenario "Vier Gewinnt auf Mobile-Phone spielen"

Martin überlegt sich welche Fachhochschule für ihn wohl die beste ist und ob er überhaupt gerade nach seiner Berufslehre mit der Fachhochschule beginnen soll. In seiner Berufsmaturitätsschule wird er informiert, dass am Samstag ein Informationstag der Hochschule Rapperswil stattfinden wird. Dies kommt für Martin wie gerufen und er entscheidet sich sofort am Informationstag teilzunehmen.

Am Informationstag fällt Martin ein Stand auf, an dem ein Roboter Vier Gewinnt spielt. Im Gespräch mit einem HSR-Student wird Martin das Projekt genauer erklärt und er wird ermuntert, den Roboter doch selber mit seinem Handy zu steuern. Er besucht die angegebene Website und kann dort einfach ein neues Spiel starten und gegen den Roboter antreten.

Erfreut über die interaktive Darstellung von Technologie kommt er mit dem HSR-Studenten noch weiter ins Gespräch und bekommt einen guten Eindruck von der Hochschule Rapperswil.

Martin kann sich nun gut vorstellen in Rapperswil ein Studium zu beginnen, da er das Gefühl hat, dass hier praktisch gearbeitet wird und auch etwas umgesetzt wird. Er will auf jeden Fall ein Studium besuchen, bei dem man nicht nur Theorie lernt. Der Infotag hat ihm gezeigt, dass dies in Rapperswil nicht der Fall ist.

6.8.2 Maximum Szenario "Vier Gewinnt am Objekt spielen"

Laura will nächstes Jahr Landschaftsarchitektur studieren. Um sich über das Studium zu informieren geht sie an den Informationstag an der Hochschule Rapperswil. An diesem Tag haben auch die anderen Fachrichtungen Informationstag. Darum werden zur Veranschaulichung Arbeiten und Projekte ausgestellt. Darunter auch ein Vier Gewinnt, welches Informatiker, Maschinentechnik und Elektrotechniker zusammen erstellt haben. Da Laura in ihrer Kindheit viel mit ihrer Schwester Vier Gewinnt gespielt hat, ist sie interessiert, schaut sich den Roboter an und will sofort selber spielen. Als Laura an der Reihe ist schaut sie sich die Bedienoberfläche des Touchscreens an. Für Laura ist es wichtig, dass die Bedienung einfach und intuitiv ist, da sie weder ein Mobile-Telefon mit Touchscreen besitzt, noch in ihrer Freizeit Videospiele spielt.

Da das Vier Gewinnt stark dem ihr bekannten Vier Gewinnt aus der Realität ähnelt, hat Laura keine Mühe sich zurecht zu finden und ihr macht es Spass zu spielen. Als sie das erste Spiel ohne Probleme gewinnt, schaut sie, ob es möglich ist den Schwierigkeitsgrad einzustellen. Dies findet sie auf Anhieb heraus und stellt ihn auf die mittlere Schwierigkeitsstufe. Nun bemerkt sie ein Symbol, mit welchem man auch mit anderen zusammen ein Multiplayerspiel gegen den Roboter machen kann. Sie berührt dieses und sofort startet ein neues Spiel mit Sebastian Spontaneous, welcher sich zur gleichen Zeit über das Mobiltelefon mit dem Spiel verbunden hat.

Nachdem sie ein paar mal gespielt hat, kommt ihr plötzlich in den Sinn, weshalb sie eigenltich hier ist und geht zu den Landschaftarchitekten um sich über ihr Studium zu informieren. Bevor sie nach Hause geht, will sie sicher noch

einmal Vier Gewinnt spielen und diesmal im schwierigsten Level.

6.8.3 Szenario "Aufbau der Vitrine"

Nachdem das Projekt "Vier Gewinnt" erfolgreich abgeschlossen wurde, will die HSR das Projekt an den Infoveranstaltungen an verschiedenen Berufsschulen präsentieren. Für diesen Zweck transportiert der Informationsdienst die Vitrine mit dem Roboter und dem Server (nachfolgend Vitrine genannt) von der HSR zu der entsprechenden Berufsschule. Dort angekommen, wird die Vitrine an den Strom angeschlossen und der Server mittels Einschaltknopf gestartet. Der Webserver und das Programm für die Spiellogik werden durch den Autostart des Betriebssystem automatisch gestartet. Anschliessend muss der Informationsdienst noch den Accesspoint einschalten. Falls die auszustellende Schule ein Internetanschluss hat, kann das Kabel nur noch in den Server eingesteckt werden.

6.9 Prototypen

Zur Veranschaulichung sind unten zwei Skizzen.

6.9.1 Schematische Übersicht der Vitrine

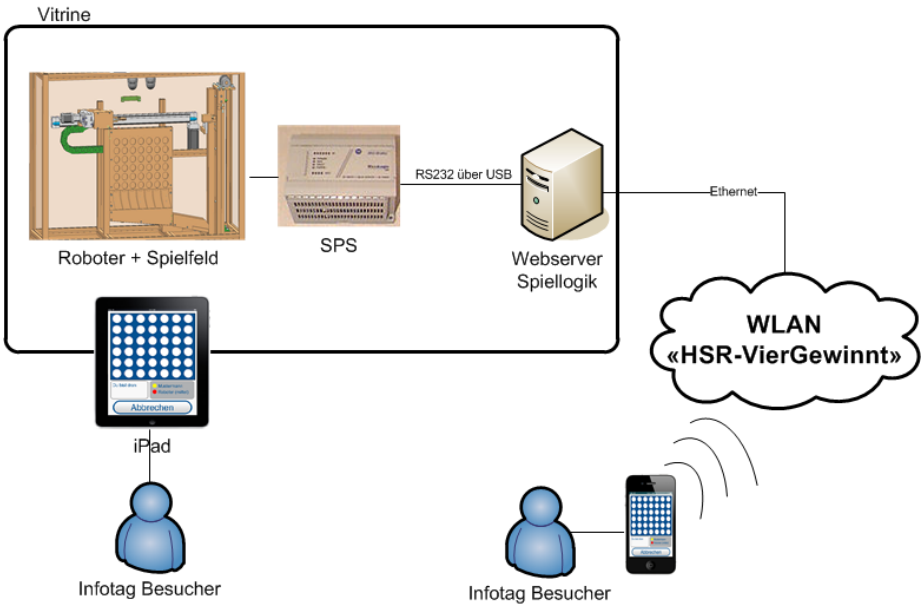


Abbildung 4: Schematische Übersicht

6.9.2 iPhone Oberfläche

Dies ist eine erste Möglichkeit wie die Benutzeroberfläche aussehen wird. Die Weiterentwicklung dieses Prototyps finden Sie im Kapitel 13 UI Design

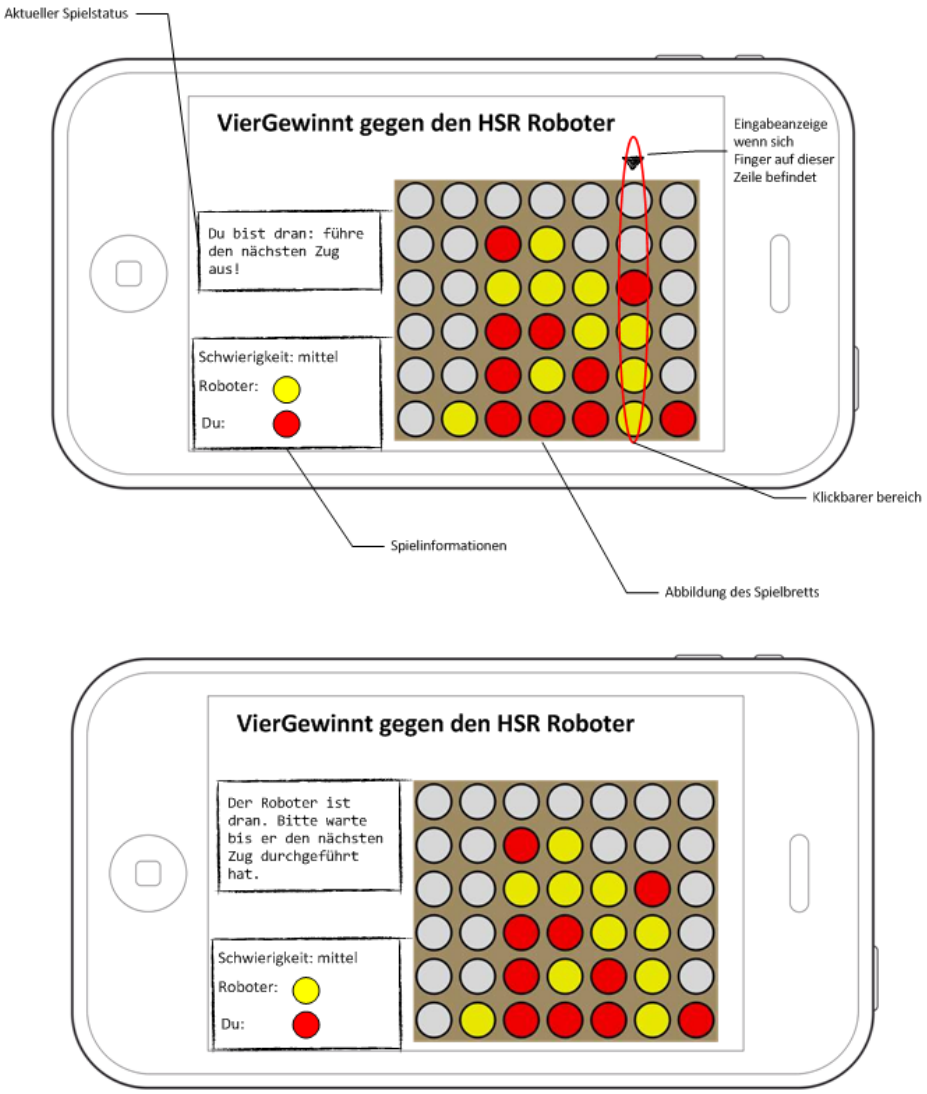


Abbildung 5: iPhone Paperprototyp Version 1

7 Zeitplanung Sprintplanung

Gemäss Vorgabe müssen pro ECTS Punkt 30 Arbeitsstunden geleistet werden. Die Semesterarbeit wird mit 8 ECTS Punkten belohnt, was 240 Arbeitsstunden entspricht. Da für die Semesterarbeit 14 Wochen zur Verfügung stehen, muss jeder Student im Durchschnitt 17 Stunden pro Woche arbeiten, d. h. 34 Stunden für beide Teammitglieder (die Gelbe Linie in der untenstehenden Grafik).

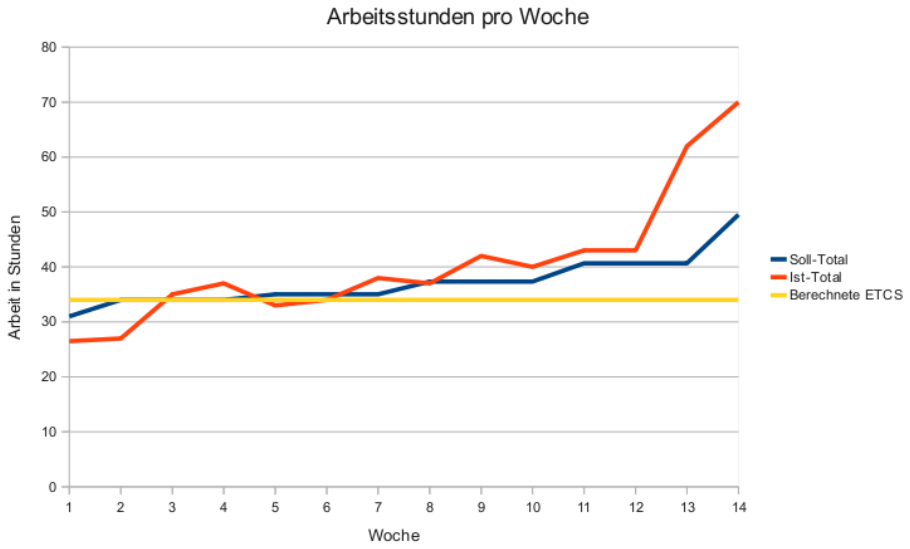


Abbildung 6: Arbeitsstunden pro Woche

7.1 Arbeitspakete

Wir haben unsere Arbeit in Pakete gegliedert. Die Namen der Pakete sind bis auf Administration selbsterklärend. In dem Arbeitspaket "Administration" haben wir zum einen organisatorische Aufgaben, wie z. B. Meetings, Meetingsprotokolle schreiben, Wiki einträge, etc. und zum anderen auch technische Aufgabe wie z. B. SVN aufsetzen, Entwicklungumgebung installieren, Accesspoint flashen, etc. reingenommen.

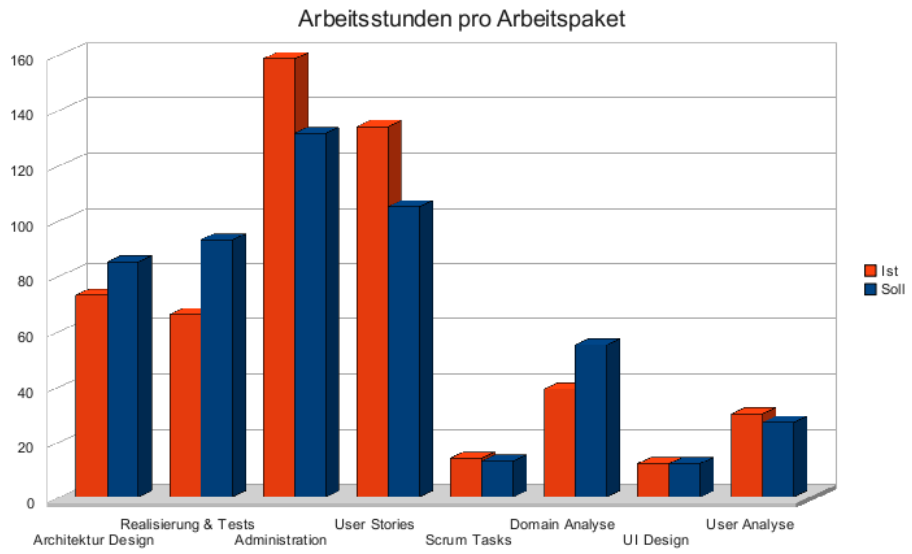


Abbildung 7: Arbeitsstunden pro Arbeitspaket

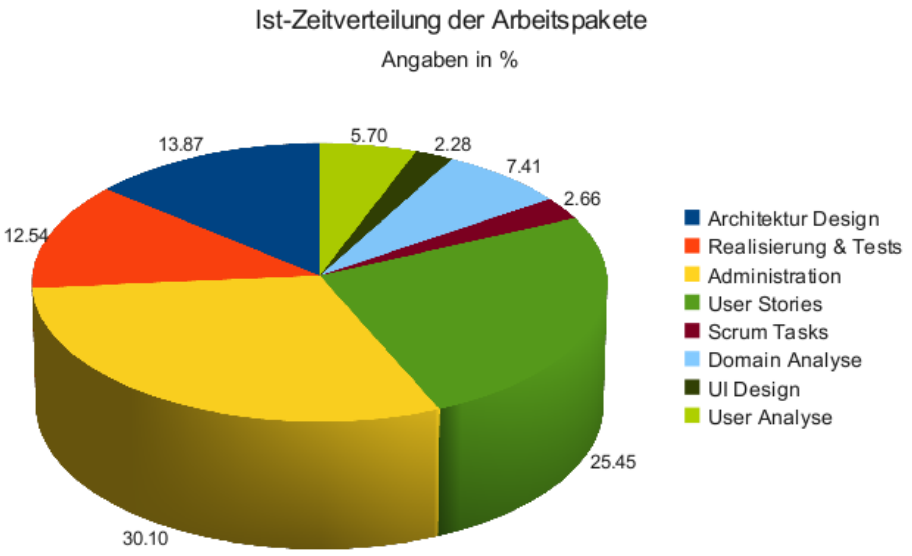


Abbildung 8: Ist-Arbeit der Arbeitspakete

7.2 Meilensteine

Wir haben die Meilensteine jeweils auf das Ende der einzelnen Sprints gelegt. Die Detailansicht der Meilensteine und deren Aufwand sind im Anhang 21 Zeit-

erfassung ersichtlich.

8 Risikomanagement

Das folgende Kapitel dient der Analyse der potentiellen Risiken und deren Vorbeugung. Es wird analysiert, welche Auswirkungen ein Eintreten eines Risikos mit sich bringt und mit welchen Massnahmen diese Risiken eingedämmt werden können.

8.1 Risikoliste

Risk ID	Risiko	Auswirkung	Massnahme	Kosten der Massnahmen In h	Max. Schaden in h	Wahrscheinlichkeit des Eintreffens	Gewichteter Schaden in H	Priorität
R01	Implementation des Minimax Spielalgorithmus ist komplizierter als Erwartet	Die Implementation dauert länger als Geplant	Zeitreserve einplanen	0	8	0.1	0.8	niedrig
R02	Die RS232 over USB Verbindung zum Roboter funktioniert nicht oder braucht Länger in der Implementation	Es müssen alternativen in Betracht gezogen werden und mit den Anderen Abteilungen Abgesprochen werden.	Früh mit der Schnittstelle beginnen und Zeitreserven einplanen	0	16	0.2	3.2	mittel
R03	Der Support von HTML5 in der Entwicklungsumgebung ist nicht gut Genug.	Die Implementation wird komplizierter und man muss sich eventuell in neue Tools Einarbeiten	Evaluation der Entwicklungsumgebung	4	16	0.15	2.4	mittel
R04	Die benötigte Hardware Wird zu spät geliefert.	Die Tests und die Anbindung an den Roboter Verzögert sich.	Genug früh die passende Hardware finden und dem Institut genug Zeit für die Beschaffung geben.	0	5	0.1	0.5	niedrig
R05	Die Kommunikation mit den anderen Abteilungen funktioniert nicht wie Gewünscht.	Die Schnittstellen können nicht sauber definiert Werden, es passieren Fehler und es gibt Verzögerungen.	Interdisziplinäres Gesamtmeeting sowie verschiedene Meetings mit den Anderen Studenten.	10	10	0.1	1	niedrig
R06	Die anderen Abteilungen erreichen ihre Ziele nicht, nur ungenügend Oder zu spät.	Das Endresultat hat nicht die gewünschte Qualität.	Unsere Teile des Gesamtprojekts mit Mockups Testen.	10	20	0.1	2	mittel
R07	Unzureichende Dokumentation der Technologien	Längere Einarbeitungszeit	Weitere Dokumentationen, Beispiele suchen und Studieren. Bücher bestellen.	0	10	0.2	2	mittel
R08	Ausfall eines Teammit-Glieds infolge Krankheit, Unfall	Verzögerung im Projekt-Verlauf	Zeitreserve einplanen	0	30	0.1	3	mittel

R09	Wissen konzentriert auf Einzelne Person	Einzelne Aufgaben können nicht erfüllt werden	Wissen weitergeben in Sitzungen	5	10	0.1	1 niedrig
R10	Probleme mit Konfigurationsverwaltung	Teile der Arbeit gehen Verloren.	Einsatz von Versionierungssystem	5	10	0.1	1 niedrig
R11	Kurz vor Ende eines Sprints wird ein grösserer Bug gefunden	Einzelne Ziele des Sprints können nicht erreicht Werden und müssen verschoben werden.	Mit Unit Tests das Risiko von Bugs vermindern. Mit Mockobjects teile des Systems Simulieren um Andere Teile zu testen.	30	20	0.3	6 hoch
R12	Hardware Defekt eines Teammitglieds	Hardware muss ersetzt werden und Konfiguration muss neu durchgeführt Werden. Möglicher Datenverlust	Alle wichtigen Daten werden ins SVN geladen.	0	20	0.05	1 niedrig
R13	Meinungsverschiedenheiten im Team führen Zu einem Streit	Produktivität und Motivation sinkt was sich auf Die Qualität der Arbeit Auswirkt.	Gutes Arbeitsklima fördern.	0	15	0.05	0.75 niedrig
R14	Anforderungen des Projekts ändern sich	Überarbeitung von Projektdokumenten	Regelmässige Sitzungen, Zeitreserve einplanen. Einsatz von Scrum	10	10	0.2	2 hoch

8.2 Massnahmen zur Risikovermeidung

Um die Risiken unter Kontrolle zu behalten und so gut wie möglich zu eliminieren, haben wir folgende allgemeinen Massnahmen definiert:

- An jedem Daily Scrum erläutert jeder Entwickler seinen aktuellen Entwicklungsstand und welche Probleme ihn gerade beschäftigen. Entsprechende Massnahmen können direkt mit dem Scrum Master besprochen werden. Somit werden zumindest die entwicklungsspezifischen Risiken abgedeckt.
- Es werden regelmässige Reviews von Programmcodes und Dokumenten durchgeführt. Jeder Code und jedes Dokument muss mindestens einmal von einer anderen Person überprüft und wenn nötig angepasst werden.

Zusätzlich zeigt die oben gezeigte Tabelle, welche konkreten Massnahmen im Bezug auf die einzelnen Risiken getroffen wurden.

9 Qualitätsmassnahmen

Im folgenden Teil werden alle qualitätssichernden Massnahmen detailliert aufgeführt. Das Ziel dieses Kapitels liegt darin, schon zum Beginn des Projektes Massnahmen zu treffen, welche ein qualitativ hochstehendes Endprodukt garantieren.

9.1 Teammeetings

Wie es der Scrum Prozess vorschlägt, halten wir mit dem ganzen Scrum Team und dem Scrum Master, wobei hier der Scrum Master einer aus dem Team ist, sogenannte „Daily Scrum“ Meetings. So kann sichergestellt werden, dass jeder den Überblick über den Gesamtfortschritt behalten kann und dass bei allfälligen Problemen direkt Massnahmen getroffen werden können um das Problem zu beheben.

Des weiteren werden wöchentlich Meetings mit unserem Prof. Dr. Markus Stolze durchgeführt, bei welchem wir ihm den neusten Stand unserer Arbeit präsentieren. Vor jedem Meeting werden die Meetingstraktanden an Prof. Dr. Markus Stolze gesendet. Es wird bei jedem Meeting ein Protokoll geführt, um Beschlüsse zu protokollieren. Die Beschlüsse und die neuen Todo's werden anschliessend an Prof. Dr. Markus Stolze gesendet und ins Wiki eingetragen.

Zusätzlich findet zu Beginn jedes Sprints ein Sprint Planning Meeting statt. An diesem Meeting wird zuerst ein Review des vergangenen Sprints durchgeführt, bevor dann der nächste Sprint im Detail geplant wird.

9.2 Codierungsrichtlinien

Grundsätzlich halten wir uns an die Java Richtlinien von Sun/Oracle [url06].

9.3 Reviews

Während des gesamten Projekt werden regelmässig verschiedene Reviews durchgeführt. Diese werden protokolliert und Fehler sofort korrigiert.

9.3.1 Code-Reviews

Code Reviews werden von den Teammitgliedern regelmässig durchgeführt. Es wird im Header des Codes dokumentiert wann die Reviews durchgeführt wurden.

Bei den Code-Reviews werden folgende Punkte beachtet:

- Verständlichkeit und Lesbarkeit des Codes
- Code Smells “riechen”
- Effizienz von komplexen Algorithmen

9.3.2 Dokumenten-Reviews

Bei den Sprint Reviews werden die Dokument von unserem Prof. Dr. Markus Stolze reviewt. Die Reviews werden im Meetingsprotokoll protokolliert.

Bei den Dokumenten-Reviews werden folgende Punkte beachtet:

- Inhalt
- Konsistenz der verschiedenen Textteile
- Rechtschreibung / Grammatik

9.4 Testing

Wir führen laufend verschiedene Tests durch. Z. B. Usability-, System-, Unittest und manuelle Test durch.

9.4.1 Unittests

Um eine geringe Kopplung und eine geringe Kopplung zu garantieren werden Unittests mittels JUnit durchgeführt.

9.5 Versionierung

Für die Versionierung verwendeten wir SVN. Für Windows nutzten wir Tortoise und auf Linux RabbitVCS um die Dateien zu synchronisieren.

Teil III
Projektdokumentation

10 User Analyse

10.1 Interview

10.1.1 Person

Alter	32
Geschlecht	Männlich
Beruf	Mitarbeiter Informationsdienst, Organisation von Informationstagen
Hat schon diverse Infotage organisiert und Objekte aufgestellt	

10.1.2 Interviewdetails

Hauptbefrager	Amon Grünbaum
Schreiber	Patrick Dünser
Vertraulichkeit garantiert	ja
Ort	Büroarbeitsplatz (Nicht am Infostand)
Dauer	1 h
Weiteres	Das Interview wurde zur Nachbearbeitung aufgenommen. (Audiodatei)

10.1.3 Interview

Die gesamten Fragen und Antworten des Interviews findet man im Anhang 22 Interview Transkript.

10.1.4 Zusammenfassung des Interviews

Die Informationsdienste freuen sich auf das neue Ausstellungsobjekt und sind zu unserem Projekt sehr positiv eingestellt. Es gibt nur sehr wenige interessante Objekte die Ausgestellt werden können um den HSR Stand attraktiv zu gestalten. Die interviewte Person konnte uns alle unsere Fragen beantworten und viele Hinweise geben, auf was wir speziell achten sollten. Mit dem Interview konnten wir unser Verständnis was für Bedürfnisse vorhanden sind verbessern. Folgende Punkte sind speziell hervorzuheben:

- Ausstellungsobjekte dienen dazu, Leute an den Stand zu bringen.

- Objekt sollte von einer Person, welche keine grossen Computerkenntnisse hat, aufgestellt und in Betrieb genommen werden können.
- Das Objekt muss relativ handlich sein und die Bedienung leicht und intuintiv.

10.1.5 Interpretation / Insights

- Priorität auf Mehrspielermodus
- Handliches Objekt (nicht zu gross)
  - Objekt evtl. zweiteilig: unterer Teil mit Rollen und oben der Roboter

10.2 Fragebogen Infotag

10.2.1 Ziel

Mit der Benutzerbefragung am Infotag wollen wir unsere Annahmen über das Zielpublikum validieren. Um verschiedene Benutzergruppen zu beschreiben haben wir Personas erstellt, in welchen fiktive Personen für eine ganze Benutzergruppe stehen. Die Resultate der Benutzerbefragungen fliessen in diese Personadokumente. Ein weiteres Ziel ist es mit einem Behavioural Pattern Diagramm die unterschiedlichen Eigenschaften der zukünftigen User zu erkennen. Bei der angewendeten Erhebungstechnik handelt es sich um eine quantitative Befragung der Endanwender. Bei dieser Marktforschung wird zusätzlich bei den Fokus Gruppen mit ein paar gezielten Fragen eine qualitative Umfrage gemacht bei der mehr ins Detail gegangen wird.

10.2.2 Vorgehen

Wir waren am Samstag 30. Oktober 2010 beim Infotag an der HSR und haben 20 Infotag-Besucher befragt. Zuerst stellten wir einige allgemeine Fragen zu ihrer Person, ihren Computerkenntissen und was für eine Mobiltelefon sie besitzen. Danach über das Spiel Vier Gewinnt und zum Schluss noch Detailfragen zu unserer SA. Die Detailfragen stellten wir nur denjenigen, welche ein internetfähiges Mobiltelefon besassen.

10.2.3 Fragebogen

Generelle Befragung:

Allgemeine Fragen:
Alle Ihre Aussagen werden Vertraulich behandelt.
Wie alt bist du?
Geschlecht:

☐ Männlich ☐ Weiblich
Wieso bist du hier?
☐ Informieren ☐ Einfach so
Für welche Studienrichtung interessierst du dich?
☐ Informatik ☐ Elektrotechnik ☐ Maschinentechnik ☐ Landschaftarchtiktur ☐ Raumplanung ☐ Erneuerbare Energie ☐ Bauingenieur
Was für ein Mobiltelefon besitzt du?
☐ iPhone ☐ Android Phone ☐ Windows Phone
In wieviel verschiedene WLAN's hast du dich letzte Woche eingeloggt?
Für was brauchst du WLAN?
☐ Surfen ☐ Mail lesen ☐ Spiele spielen
Wie schätzt du deine Computerkenntnisse ein?
☐ Sehr gut ☐ gut ☐ mittel ☐ eher schlecht ☐ sehr schlecht
Hast du einen Facebook Account
☐ Ja ☐ Nein
Fragen zum Vier Gewinnt Roboter:
Kennst du das Spiel Vier Gewinnt?
☐ Ja ☐ Nein
Wann hast du es zum letzten Mal Vier Gewinnt gespielt?
☐ letzte Woche ☐ vor einigen Monaten ☐ letztes Jahr ☐ in der Kindheit
Danke für deine Mithilfe am Projekt!

Detail Befragung:

Wenn es an diesem Infotag einen Vier Gewinnt Roboter geben würde, würdest du ihn ausprobieren?
☐ Ja ☐ Nein
Begründung:

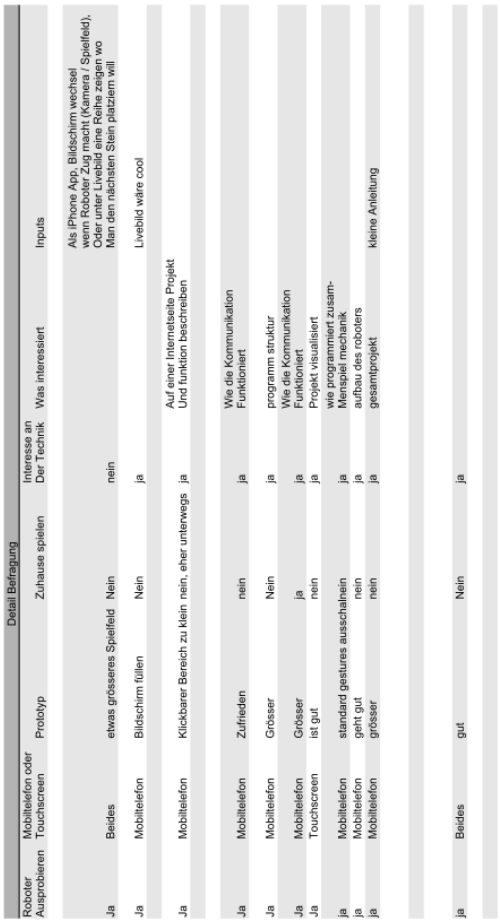
Was würdest du bevorzugen, Vier Gewinnt auf deinem eigenen Mobiltelefon oder auf einem Touchscreen am Objekt zu spielen?
☐ Mobiltelefon ☐ Touchscreen
Begründung:
(Prototyp zeigen und erklähren) Kannst du auf diesem Prototypen die Eingabe so durchführen wie du es gerne hättest?
Gehe auf die Website vierGewinnt.slackline.ch und versuche ein neues Spiel zu starten.
Führe einige züge aus.
Wie kommst du mit der Steuerung zurecht?
Würdest du von Zuhause aus auch spielen?
☐ Ja ☐ Nein
Interessierst du dich für die Technik hinter dem Ausstellungsobjekt?
☐ Ja ☐ Nein
Wenn ja, was interessiert dich dabei genau?
Sonstige Ideen und Inputs?
Danke für deine Mithilfe am Projekt!

10.2.4 Zusammenfassung

Alle Antworten der befragten Besucher haben wir in einer Tabelle zusammengefasst.

	Allgemeine Fragen										Fragen zum Vier Gewinnt (-Roboter)		
	Alter	Geschlecht	Grund des Kommens	Studienrichtung	Mobiltelefon	WLAN einloggen	Für was WLAN	Computer-Kenntnis	Facebook-Account	Kannst du Vier Gewinnt	Zum letzten Mal gespielt
1	20	Männlich	Informieren	Baungenieur	--			gut	ja	ja	kindheit
2	20	Männlich	Informieren	Elektrotechnik	iPhone	täglich	surfen, mail lesen, Spielen	mittel	ja	ja	letztes Jahr
3	21	Männlich	Informieren	Elektrotechnik, Informatik	Android Phone	2 mal	surfen, mail lesen, Informieren	sehr gut	ja	ja	kindheit
4	18	Männlich	Informieren	Informatik	--			gut	ja	ja	letzte Woche
5	26	Männlich	Informieren	Erneuerbare Energie	iPhone	selten	surfen	mittel	ja	ja	4 Jahre
6	32	Männlich	Informieren	Landschaftsarchitektur	--			mittel	nein	ja	kindheit
7	19	Männlich	Informieren	Landschaftsarchitektur	--			gut	ja	ja	letztes Jahr
8	20	Männlich	Informieren	Erneuerbare Energie	iPhone	4 mal mit offenen wird	surfen, mail lesen, Spielen	mittel	ja	ja	vor einigen Monaten
9	20	Männlich	Informieren	Maschinentechnik	Android Phone	Automatisch verbunden	surfen, spielen	sehr gut	ja	ja	letztes Jahr
10	20	Weiblich	Informieren	Erneuerbare Energie	iPod	selten	surfen	gut	ja	ja	kindheit
11	23	Männlich	Informieren	Maschinentechnik	--			gut	ja	ja	letztes Jahr
12	19	Männlich	Informieren	Informatik	Android Phone	selten	surfen, mail lesen	sehr gut	ja	ja	kindheit
13	20	Männlich	Informieren	Landschaftsarchitektur	iPhone	selten	surfen, informieren	eher schlecht	ja	ja	letztes Jahr
14	19	Männlich	Informieren	Erneuerbare Energie	iPhone	viel	mail lesen	sehr gut	ja	ja	letztes Jahr
15	20	Männlich	Informieren	Landschaftsarchitektur, Raumplannung	--			mittel	ja	ja	kindheit
16	20	Männlich	Informieren	Landschaftsarchitektur	--			mittel	ja	ja	kindheit
17	20	Männlich	Informieren	Raumplanung, Baungenieur	--			mittel	ja	ja	2 Jahren
18	23	Männlich	Informieren	Maschinentechnik	iPhone	ab und zu	surfen, mail lesen	mittel	ja	ja	letztes Jahr
19	19	Männlich	Informieren	Maschinentechnik	--			mittel	ja	ja	letztes Jahr
20	21	Männlich	Informieren	Informatik	--			sehr gut	ja	ja	letztes Jahr

Abbildung 9: Auswertung Fragebogen Teil 1



Detail Befragung								
Roboter Ausprobieren	Mobiltelefon oder Touchscreen	Prototyp	Zuhause spielen	Interesse an Der Technik	Was interessiert	Inputs
Ja	Beides	etwas grösseres Spielfeld	Nein	nein		Als iPhone App, Bildschirm wechsel wenn Roboter Zug macht (Kamera / Spielfeld). Oder unter Livebild eine Reihe zeigen wo Man den nächsten Stein platziern will
Ja	Mobiltelefon	Bildschirm füllen	Nein	ja		Livebild wäre cool
Ja	Mobiltelefon	Klickbarer Bereich zu klein	nein, eher unterwegs	ja	Auf einer Internetseite Projekt Und funktion beschreiben	
Ja	Mobiltelefon	Zufrieden	nein	ja	Wie die Kommunikation Funktioniert	
Ja	Mobiltelefon	Grösser	Nein	ja	programm struktur	
Ja	Mobiltelefon	Grösser	ja	ja	Wie die Kommunikation Funktioniert	
Ja	Touchscreen	ist gut	nein	ja	Projekt visualisiert	
ja	Mobiltelefon	standard gestures ausschalnein		ja	wie programmiert zusam- Menspiel mechanik	
ja	Mobiltelefon	geht gut	nein	ja	aufbau des roboters	
ja	Mobiltelefon	grösser	nein	ja	gesamtprojekt	kleine Anleitung
ja	Beides	gut	Nein	ja		

Abbildung 10: Auswertung Fragebogen Teil 2

Allgemeiner Teil

Der durchschnittliche Infotagbesucher ist 21 Jahre alt, männlich, besitzt einen Facebook-Account und ist am Infotag weil er im Sommer das Studium beginnen will. Die Interessen für die Studienrichtung sind sehr unterschiedlich, aber doch ausgeglichen. Die Hälfte der Besucher besitzen ein internetfähiges Mobiltelefon. Es lässt sich allerdings nicht festellten, ob es eine korrelation zwischen internetfähigen Mobiltelefonen und der Studienrichtung gibt. Die meisten Besucher, die ein internetfähiges Mobiltelefon besitzen gehen nur in offene WLAN, mit welchen sie automatisch verbunden werden. In andere WLAN's verbindet sich der Standard-Besucher nur selten. Das WLAN wird meist fürs surfen und Mail lesen benutzt.

Die Besucher schätzen ihre Computerkenntnisse mittel bis gut ein. D. h. sie sind geübt im Umgang mit PC's und können ohne probleme Programme

installieren und die normalen Probleme lösen. Das Spiel Vier Gewinnt kennen alle Besucher, viele von ihnen haben es innerhalb des letzten Jahres zum letzten Mal gespielt.

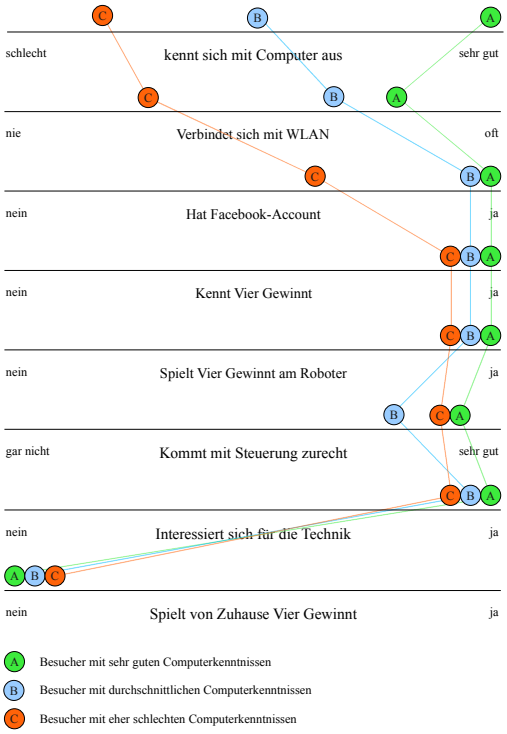
Detailierter Teil

Alle Befragten würden Vier Gewinnt gegen den Roboter spielen und bevorzugen das Spielen am eigenen Mobiltelefon. Allerdings ist diese Resultat fragwürdig, da wir die Detailbefragung nur bei Leute mit einem Mobiltelefon durchgeführt haben. Das Interesse von Zuhause gegen den Vier Gewinnt-Roboter zu spielen ist gering, so das das Ausstellungsobjekt effektiv nur als Ausstellungsobjekt genutzt werden kann. Bis auf eine Person interessieren sich alle für die Technik hinter dem Ausstellungsobjekt. Das Interesse ist sehr unterschiedlich d. h. von der Programmierspache über die Kommunikation zwischen Server und Roboter bis hin zum Aufbau des Roboters.

10.2.5 Interpretation / Insights

- Spielfeld muss grösser sein (Bildschirmfüllend)
- Horizontaler Abstand zwischen den Spalten bisschen grösser, damit der klickbare Bereich grösser wird
- Umschalten zwischen Zugeingabe und dem Ausführen des Zuges durch den Roboter
- Livestream der Vitrine auf dem Mobiltelefon und darunter eine Reihe mit 7 Knöpfen, welche die möglichen Einwurfmöglichkeiten darstellen

10.2.6 Behavioural Pattern Diagramm


schlecht
kennt sich mit Computer aus
sehr gut
nie
Verbindet sich mit WLAN
oft
nein
Hat Facebook-Account
ja
nein
Kennt Vier Gewinnt
ja
nein
Spielt Vier Gewinnt am Roboter
ja
gar nicht
Kommt mit Steuerung zurecht
sehr gut
nein
Interessiert sich für die Technik
ja
nein
Spielt von Zuhause Vier Gewinnt
ja
A Besucher mit sehr guten Computerkenntnissen
B Besucher mit durchschnittlichen Computerkenntnissen
C Besucher mit eher schlechten Computerkenntnissen

10.3 Personas

Die Personas sind eine Kombination aus den Umfrageergebnissen und unserer vorherigen Analyse.

10.3.1 Laura Landscape

Rolle: Mässig erfahrener Benutzer

Abbildung 11: Mässig erfahrener Benutzer

- 20 Jahre alt
- Gärtner Lehre mit BMS
- Will Landschaftsarchitektur studieren
- Durchschnittliche Computerkenntisse

Laura Landscape hat eine Lehre als Gärtner absolviert, welche sie sehr interessant fand und ihr gut gefallen hat. Deshalb will Laura ihr Wissen auf diesem Gebiet vertiefen und informiert sich über ein Bachelorstudium für Landschaftsarchitektur. Während der Lehre hat Laura nebenbei einen Computerkurse besucht, bei dem sie die nötigsten Fähigkeiten kennengelernt hat. Sie lernte zum Beispiel wie man Dokumente bearbeitet, Powerpoint-Präsentation erstellet und sich beim Surfen im Internet richtig verhaltet. Allerdings hat sie Computer und Technik im allgemeinen nie sehr interessiert. Sie braucht den PC um E-Mails abzurufen, über Facebook Statusupdates zu machen und zwischendurch einmal um etwas auf Wikipedia nachzulesen. Das Mobil-Telefon benützt sie zum SMS schreiben und telefonieren, deshalb hat sie nur ein altes Nokia ohne Touchscreen.

10.3.2 Martin Mobile

Rolle: Erfahrener Benutzer

Abbildung 12: Erfahrener Benutzer

- 19 Jahre alt
- Informatik Lehre mit BMS
- Gute PC-Kenntnisse
- Interessiert an aktuellen Technologien

Martin Mobil ist gerade im Abschlussstress seiner Berufslehre als Informatiker mit Schwerpunkt Applikationsentwicklung. Martin arbeitet bei einer Grossbank im Raum Zürich und geht daneben einen Tag in die technische Berufsschule und einen Tag in die Berufsmaturitätsschule. Er hat diesen anstrengenden Weg ausgewählt, um sich nach seiner Lehre alle Möglichkeiten für Weiterbildungen offen zu halten.

In der Grossbank ist er in einem kleinen Entwicklungsteam dabei mit Java eine unkritische Erweiterung im lokalen Buchhaltungssystem der Abteilung zu realisieren.

Neben Arbeit und Schulabschluss ist für Martin auch die Planung seiner Zukunft wichtig. Gerne würde er bald ein Studium beginnen. Er sagt: "Die Lehre hat mir viele Möglichkeiten der Informatik gezeigt, aber ich möchte mich nun noch mehr in gewissen Themen vertiefen."

Auch ausserhalb der Arbeit ist Martin sehr Technologie interessiert. Beim iPhone 3G war er einer der ersten in der Schweiz der eines besass und nun hat er auch schon auf ein Android Phone gewechselt. Das mobile Internet und dessen Möglichkeiten begleiten ihn immer. Weitere Hobbies von Martin sind Computerspiele, sich mit Freunden treffen sowie Gesellschaftsspiele spielen.

10.4 Szenario

Die Szenarios sind eine Kombination aus den Umfrageergebnissen, dem Interview und unserer vorherigen Analyse.

10.4.1 Szenario "Vier Gewinnt am Objekt spielen" (Maximum Szenario)

Laura will nächstes Jahr Landschaftsarchitektur studieren. Um sich über das Studium zu informieren geht sie an den Informationstag an der Hochschule Rapperswil. An diesem Tag haben auch die anderen Fachrichtungen Informationstag. Darum werden zur Veranschaulichung Arbeiten und Projekte ausgestellt. Darunter auch ein Vier Gewinnt, welches Informatiker, Maschinentechnik und Elektrotechniker zusammen erstellt haben. Da Laura in ihrer Kindheit viel mit ihrer Schwester Vier Gewinnt gespielt hat, ist sie interessiert, schaut sich den Roboter an und will sofort selber spielen. Als Laura an der Reihe ist schaut sie sich die Bedienoberfläche des Touchscreens an. Für Laura ist es wichtig, dass die Bedienung einfach und intuitiv ist, da sie weder ein Mobile-Telefon mit Touchscreen besitzt, noch in ihrer Freizeit Videospiele spielt.

Da das Vier Gewinnt stark dem ihr bekannten Vier Gewinnt aus der Realität ähnelt, hat Laura keine Mühe sich zurecht zu finden und ihr macht es Spass zu spielen. Als sie das erste Spiel ohne Probleme gewinnt, schaut sie, ob es möglich ist den Schwierigkeitsgrad einzustellen. Dies findet sie auf Anhieb heraus und stellt ihn auf die mittlere Schwierigkeitsstufe. Nun bemerkt sie ein Symbol, mit welchem man auch mit anderen zusammen ein Multiplayerspiel gegen den Roboter machen kann. Sie berührt dieses und sofort startet ein neues Spiel mit Sebastian Spontaneous, welcher sich zur gleichen Zeit über das Mobiltelefon mit dem Spiel verbunden hat.

Nachdem sie ein paar mal gespielt hat, kommt ihr plötzlich in den Sinn, weshalb sie eigenltich hier ist und geht zu den Landschaftarchitekten um sich über ihr Studium zu informieren. Bevor sie nach Hause geht, will sie sicher noch einmal Vier Gewinnt spielen und diesmal im schwierigsten Level.

10.4.2 Szenario "Vier Gewinnt auf Mobile-Phone spielen" (Minimum Szenario)

Martin überlegt sich welche Fachhochschule für ihn wohl die beste ist und ob er überhaupt gerade nach seiner Berufslehre mit der Fachhochschule beginnen soll. In seiner Berufsmaturitätsschule wird er informiert, dass am Samstag ein Informationstag der Hochschule Rapperswil stattfinden wird. Dies kommt für Martin wie gerufen und er entscheidet sich sofort am Informationstag teilzunehmen.

Am Informationstag fällt Martin ein Stand auf, an dem ein Roboter Vier Gewinnt spielt. Im Gespräch mit einem HSR-Student wird Martin das Projekt genauer erklärt und er wird ermuntert, den Roboter doch selber mit seinem Handy zu steuern. Er besucht die angegebene Website und kann dort einfach ein neues Spiel starten und gegen den Roboter antreten.

Erfreut über die interaktive Darstellung von Technologie kommt er mit dem HSR-Studenten noch weiter ins Gespräch und bekommt einen guten Eindruck von der Hochschule Rapperswil.

Martin kann sich nun gut vorstellen in Rapperswil ein Studium zu beginnen, da er das Gefühl hat, dass hier praktisch gearbeitet wird und auch etwas umgesetzt wird. Er will auf jeden Fall ein Studium besuchen, bei dem man nicht nur

Theorie lernt. Der Infotag hat ihm gezeigt, dass dies in Rapperswil nicht der Fall ist.

10.4.3 Szenario "Aufbau der Vitrine"

Nachdem das Projekt "Vier Gewinnt" erfolgreich abgeschlossen wurde, will die HSR das Projekt an den Infoveranstaltungen an verschiedenen Berufsschulen präsentieren. Für diesen Zweck transportiert der Informationsdienst die Vitrine mit dem Roboter und dem Server (nachfolgend Vitrine genannt) von der HSR zu der entsprechenden Berufsschule. Dort angekommen, wird die Vitrine an den Strom angeschlossen und der Server mittels Einschaltknopf gestartet. Der Webserver und das Programm für die Spiellogik werden durch den Autostart des Betriebssystem automatisch gestartet. Anschliessend muss der Informationsdienst noch den Accesspoint einschalten. Falls die auszustellende Schule ein Internetanschluss hat, kann das Kabel nur noch in den Server eingesteckt werden.

11 Feature Liste/Product Backlog

Während dem Projekt gesammelte Ideen wurden nach dem Projektende vom Product Backlog in eine Liste mit optionalen Features transferiert.

11.1 Optionale Features

Dieses Dokument beschreibt optionale Features, welche wir und Prof. Dr. Markus Stolze während userer Semesterarbeit angedacht haben. Bei gewissen Features reichte die Zeit nicht für eine verwirklichung, andere haben wir nicht implementiert, da wir von ihnen nicht überzeugt waren. Alle diese Features könnten in einer zusätzlichen Arbeit realisiert werden.

Nr.	Feature	Priorität
1	Demokratischer Abstimmungsmodus	Hoch
2	Server hat Internetverbindung	Mittel
3	Kamera an der Vitrine	Niedrig
4	Zufallelement (Hard- / Software)	Mittel
5	Mimik von Roboter	Niedrig
6	Laufschriftanzeige an der Vitrine	Mittel
7	Anzeige des Algorithmus	Mittel
8	Facebook integration	Niedrig

Tabelle 6: Optionale Features

11.1.1 Demokratischer Abstimmungsmodus

Ein Spielmodus, bei welchem mehrere Leute in einem Team sind und über eine demokratische Abstimmung wählen, wo der nächste Stein eingeworfen werden kann.

11.1.2 Server hat Internetverbindung

Server besitzt eine Verbindung ins Internet, so dass man auch von "Aussen", also nicht nur im WLAN "HSR-VierGewinnt", z.B. über die Adresse "viergewinnt.hsr.ch" gegen den Roboter spielen kann. Dafür müsste man sich mit dem IT-Desk in Verbindung setzen, um die Adresse reservieren zu lassen.

11.1.3 Kamera an der Vitrine

Um die im Kapitel 11.1.2 beschriebene Erweiterung attraktiver zu machen, kann eine Kamera an der Vitrine montiert werden, welche das Spielfeld und den Roboter filmt. Damit würden man sehen, das man nicht nur ein gewöhnliches Vier Gewinnt spielt, sondern die Steine durch einen Roboter eingefügt werden.

11.1.4 Zufallelement (Hard- / Software)

Ein Zufallselement einbauen. Hier gibt es zwei Möglichkeiten. Zum einen, dass das Element hardwaremässig implementiert wird. Zum anderen softwaremässig.

- Hardware: Ein Nagelbrett über dem Spielfeld, durch welches der Stein seinen weg in eine Spalte finden muss.
- Software: Durch einen Random-Generator wird nicht immer die optimale Spalte gewählt, sondern eine links oder rechts davon.

11.1.5 Mimik von Roboter

Der Roboter könnte hämisch lachen, wenn er den gewinnbringenden Zug ausführt. Oder bereits, wenn der Spieler an einer nicht optimalen Stelle den Stein einwirft.

11.1.6 Laufschriftanzeige an der Vitrine

Überhalb der Vitrine wird eine Laufschrift montiert, auf welcher aktuelle Informationen zum Spiel zu lesen sind (z.B. Rot gewinnt das Spiel).

11.1.7 Anzeige des Algorithmus

Auf einem zusätzlichen Bildschirm wird der Aufbau des Search-Trees und dessen Traversierung durch den Algorithmus aufgezeigt.

11.1.8 Facebook integration

Eine Facebook-App mit der man übers Internet Vier Gewinnt am Roboter spielen kann. Eventuell mit einer Highscore welche auf dem Profile angezeigt wird.

12 Domainanalyse

12.1 Domain Model

12.1.1 Diagramm

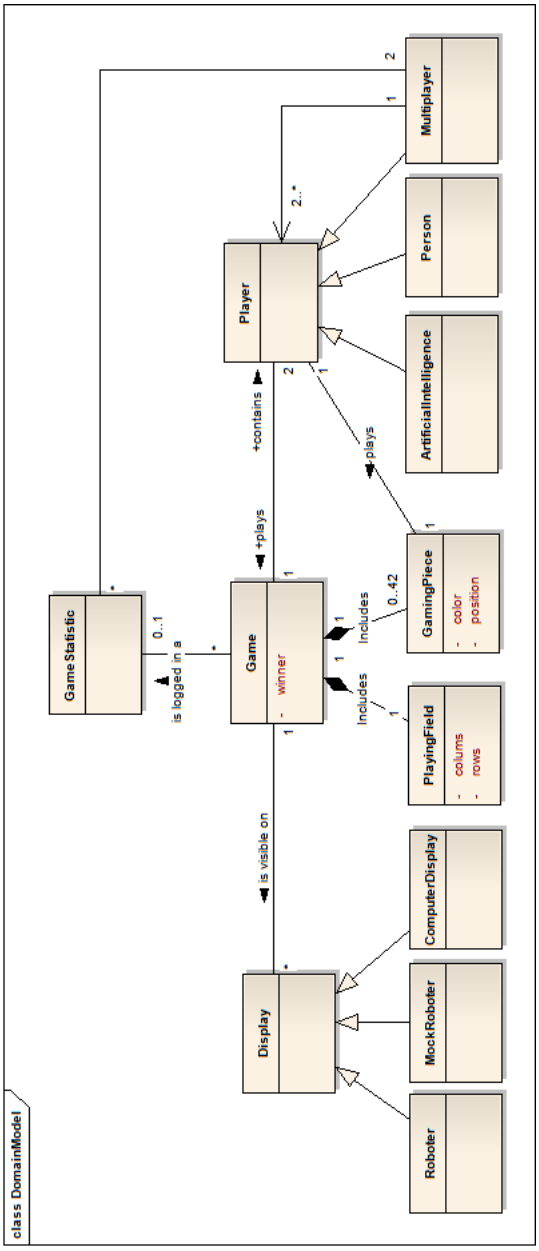


Abbildung 13: Domain Modell

12.1.2 Beschreibung

Das Domain Model ist eine visuelle Darstellung der Problemstellung unseres Programms. Es zeigt die verschiedenen Bereiche, die im Programm vorkommen und wie diese zusammenspielen. Auf der Basis dieses Modells kann später das Design Model erstellt werden und es können Softwareklassen generiert werden. Das Domain Model wird im verlauf der implementierung nicht mehr angepasst. Nachfolgend eine Beschreibung der einzelnen konzeptuellen Klassen (keine Softwareklassen):

Game

Beschreibung:	Die Klasse Game wiederspiegelt ein einzelnes Spiel, welches gerade am Laufen ist. Zu jedem Zeitpunkt darf nur ein Spiel am laufen sein.
Attribute:	winner: Enthält den Gewinner des Spiels.
Beziehung:	Ein Spiel hat immer genau zwei Spieler. Zudem gibt es ein Spielfeld. Während dem Spiel werden dem Game neue Spielsteine hinzugefügt (max. 42). Jedes Spiel kann auf beliebig vielen Displays angezeigt werden. Dazu gehört auch die Anzeige über dem Roboter. Jedes Game wird in einer Statistik eingetragen.

Display

Beschreibung:	Diese Klasse ist für die Anzeige des Spiels zuständig. Je nach Situation gibt es verschiedene Möglichkeiten für diese Klasse.
Varianten:	Roboter: Diese Klasse repräsentiert die Anzeige des Spiels durch den Roboter. ComputerDisplay: Das Spiel wird auf einem Computer angezeigt. MockRoboter: Es wird nur simuliert, dass das Spiel auf dem Roboter angezeigt wird.
Attribute:	
Beziehung:	Ein Display zeigt immer genau ein Spiel.

Player

Beschreibung:	Die Klasse Player enthält einen Spieler. Für jedes Spiel braucht es zwei Spieler.
Varianten:	Als Spieler kann ein Mensch oder auch eine künstliche Intelligenz auftreten. Zudem ist es auch möglich das in einem Multiplayer Spiel mehrere Menschen zusammmen den Platz eines Spielers einnehmen.
Attribute:	
Beziehung:	In jedem Spiel braucht es genau zwei Spieler. In jedem Zug spielt ein Spieler einen Spielstein.

GameStatistic

Beschreibung:	In der GameStatistic werden die Spiele aufgezeichnet. Mit diesen Informationen ist es möglich eine Statistik anzuzeigen, die zum Beispiel die Anzahl Siege des Roboters mit der Anzahl Siegen von Menschen vergleicht. Weiter ist es über die Game Statistik möglich alte Spiele nochmals laufen zu lassen, falls gerade kein neues Spiel gespielt wird.
Attribute:	
Beziehung:	Jedes Spiel sollte in der GameStatistic aufgezeichnet werden.

PlayingField

Beschreibung:	Die Klasse PlayingField repräsentiert das Spielfeld.
Attribute:	Die Zeilen und Spalten des Spielfelds.
Beziehung:	In jedem Game gibt es genau ein Spielfeld.

GamingPiece

Beschreibung:	Das GamingPiece ist ein einzelner Spielstein. Dieser wird erstellt und von einem Spieler ins Spielfeld gelegt.
Attribute:	Ein Spielstein hat entweder die Farbe Rot oder Gelb. Sobald er gespielt wurde, hat er auch noch eine Position im Spielfeld.
Beziehung:	Ein Spielstein wird von einem Spieler gespielt und gehört zu einem Spiel.

12.2 Use Case

12.2.1 Use Case Diagramm

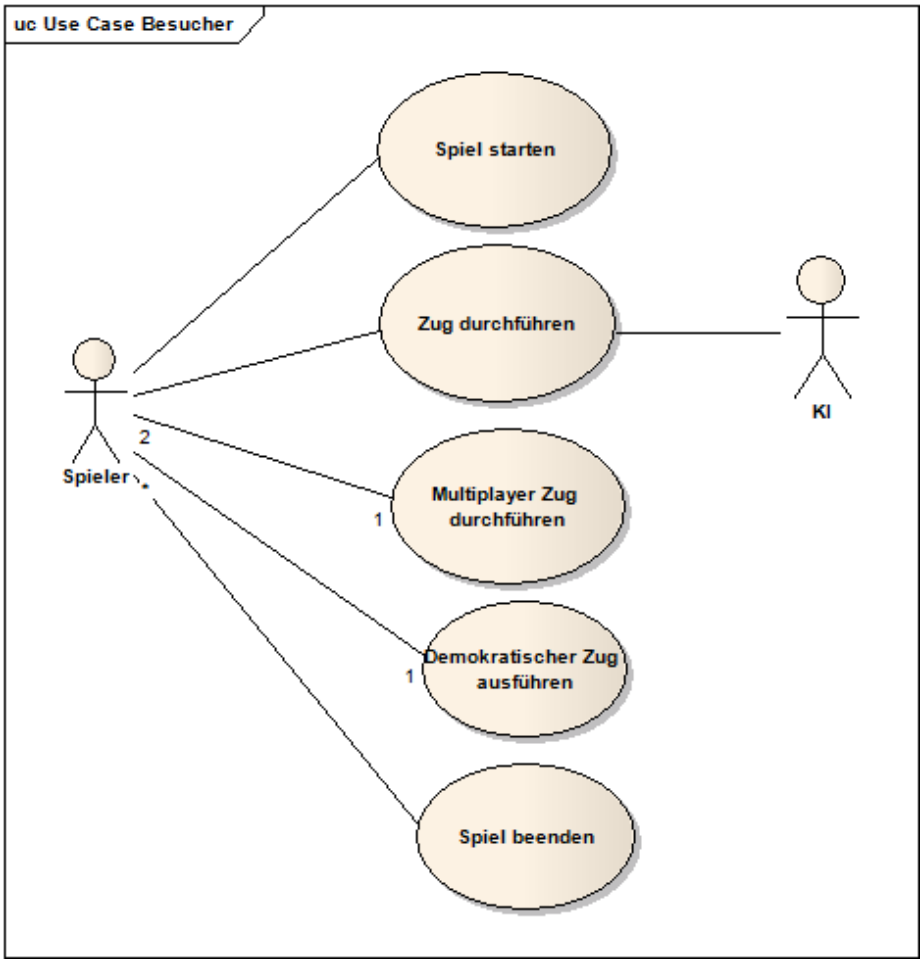


Abbildung 14: Besucher Use Case Diagramm

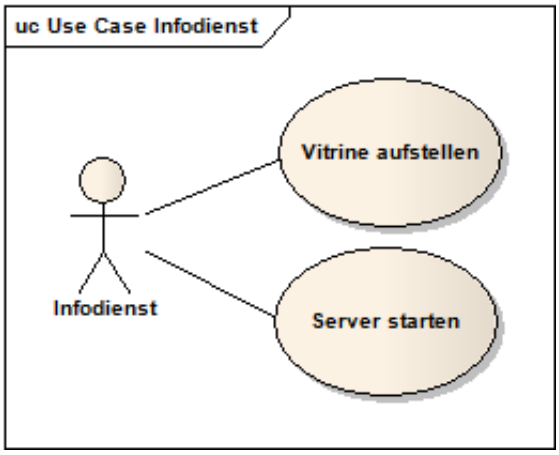


Abbildung 15: Infodienst Use Case Diagramm

12.2.2 Aktoren & Stakeholder

Es gibt die Aktoren User und KI, welche den Informationstag-Besucher respektive die Spiellogik auf dem Server, kurz künstliche Intelligenz, representieren.

12.2.3 Spiel starten

Overview	Benutzer startet Spiel
Stakeholders & Interests	Benutzer: Möchte Vier Gewinnt spielen
Preconditions	• Server muss laufen • Benutzer muss Verbindung zu einem WLAN haben • Browser muss die verwendete Websprache unterstützen
Success Guarantee	Benutzer ist mit Server verbunden und kann Optionen im Menü wählen

Main Success Scenario	1. Grundinformationen anfordern 2. Schwierigkeit eingeben 3. Spielmodus eingeben 4. Wählt Menü-Punkt “Spiel starten”
Extension / Alternative Flows	• 1. a) Browser unterstützt die verwendete Websprache nicht • 1. b) Server nicht online • 4. a) Das Spiel kann nicht gestartet werden, da jemand anderes bereits am spielen ist
Special Requirements	
Technology and Data Variation List	

12.2.4 Zug durchführen

Overview	Benutzer spielt Vier Gewinnt gegen Computergegner
Stakeholders & Interests	Benutzer: Möchte ein Zug gegen den Computergegner machen

Preconditions	• Benutzer muss ein Spiel gestartet haben und es läuft
Success Guarantee	Benutzer sieht auf dem Spielfeld das sein Zug durchgeführt wurde
Main Success Scenario	1. Benutzer klickt auf Spielfeld 2. Roboter führt dessen Zug aus 3. KI macht Zug 4. Roboter führt dessen Zug aus
Extension / Alternative Flows	1. a) Benutzer klickt auf falsche Spalte 2. a) Roboter führt Zug nicht aus
Special Requirements	
Technology and Data Variation List	
Frequency of Occurence	Benutzer: sehr oft (mehrmals pro Spiel und vielemale an einem Infotag)
Miscellaneous	

12.2.5 Multiplayer Zug durchführen

Overview	Benutzer spielt Vier Gewinnt gegen Computergegner
Stakeholders & Interests	Benutzer: Möchte ein Spiel gegen einen anderen Besucher spielen
Preconditions	• Benutzer muss Spiel gestartet haben • Gegner muss verbunden sein
Success Guarantee	Benutzer kann gegen menschlichen Gegner Vier Gewinnt spielen
Main Success Scenario	1. Benutzer klickt auf Spielfeld 2. Roboter führt dessen Zug aus 3. Menschlicher Gegner macht Zug 4. Roboter führt dessen Zug aus
Extension / Alternative Flows	1. a) Alle Spieler eines Teams können für den nächsten Zug voten
Special Requirements	
Technology and Data Variation List	

Frequency of Occurence	Benutzer: oft (mehrmals an einem Infotag)
Miscellaneous	

12.2.6 Spiel beenden

Overview	Spiel wird beendet
Stakeholders & Interests	Spiel: es wird beendet
Preconditions	• Spiel ist am laufen
Success Guarantee	Das Hauptmenü wird angezeigt
Main Success Scenario	1. Besucher oder KI hat Spiel gewonnen
Extension / Alternative Flows	• 1. a) Besucher hat Spiel abgebrochen • 1. b) Besucher hat 3 Minuten keine Eingabe gemacht

Special Requirements	Timer in der Applikation
Technology and Data Variation List	
Frequency of Occurence	Benutzer: oft (bei jedem Spiel)
Miscellaneous	

12.2.7 Vitrine aufstellen

siehe User Analyse Szenario “Aufbau der Vitrine” (Kap. 10.4.3)

12.2.8 Server starten

siehe User Analyse Szenario “Aufbau der Vitrine” (Kap. 10.4.3)

13 UI Design

13.1 Prototyp

13.1.1 1. Version

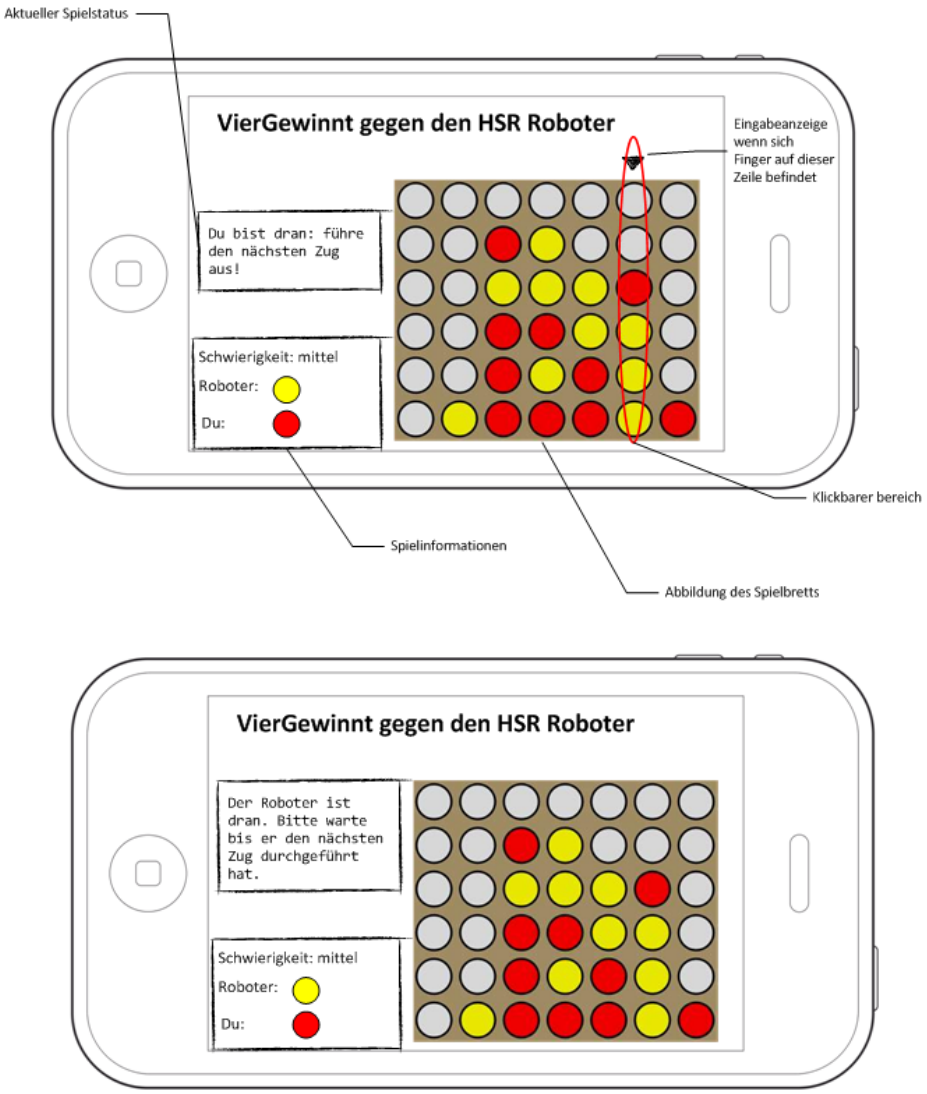


Figure 16: iPhone Prototyp 1. Version

13.1.1.1 Heuristische Evaluation

Heuristische Evaluation (für Prototyp Version 1)

Kriterien

Sichtbarkeit des System-Status

☒ sehr gut ☐ gut ☐ mässig ☐ schlecht

Enger Bezug zwischen System und realer Welt

☒ sehr gut ☐ gut ☐ mässig ☐ schlecht

Nutzerkontrolle und Freiheit

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Konsistenz & Konformität mit Standards

☐ sehr gut ☐ gut ☒ mässig ☐ schlecht

Fehler-Vorbeugung

☐ sehr gut ☐ gut ☒ mässig ☐ schlecht

Besser Sichtbarkeit als Sich-erinnern-Müssen

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Flexibilität und Nutzungseffizienz

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Ästhetik und minimalistischer Aufbau

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Insights

- Benutzer braucht einen grösseren Klickbereich
- Zwei Modus Landscape und Portrait
- Benutzer muss nicht wissen gegen welchen Schwierigkeitsgrad er spielt

Redesign Entscheide

- Spielfeld etwas grösser, dafür Titel weg
- Vertikaler Abstand zwischen Spalten grösser
- Es muss ein zweiter Modus (Portrait) eingeführt werden, bei dem die Statusanzeige unten ist
- iPhone- und Android-Standards müssen wenn möglich eingehalten werden

13.1.2 Prototyp 2. Version

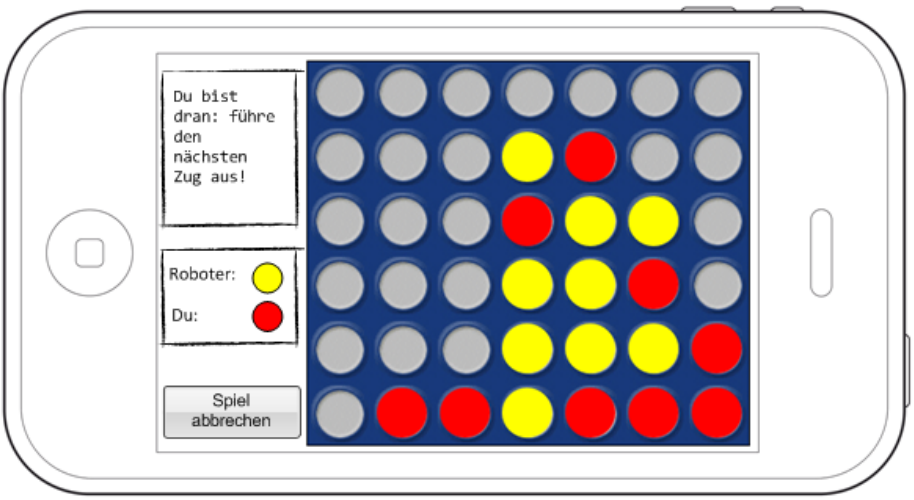


Figure 17: iPhone Prototyp 2. Version (Landscape)

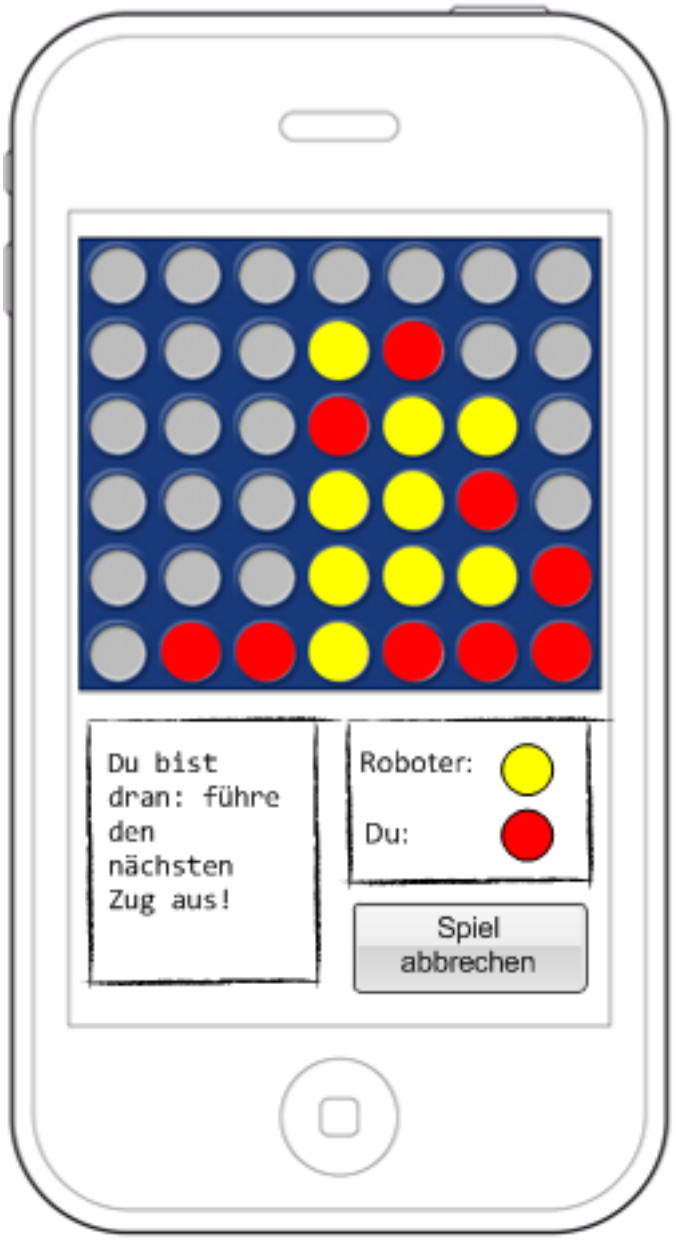


Figure 18: iPhone Prototyp 2. Version (Portrait)

13.1.2.1 Heuristische Evaluation

Heuristische Evaluation (für Prototyp Version 2)

Kriterien

Sichtbarkeit des System-Status

☒ sehr gut ☐ gut ☐ mässig ☐ schlecht

Enger Bezug zwischen System und realer Welt

☒ sehr gut ☐ gut ☐ mässig ☐ schlecht

Nutzerkontrolle und Freiheit

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Konsistenz & Konformität mit Standards

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Fehler-Vorbeugung

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Besser Sichtbarkeit als Sich-erinnern-Müssen

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Flexibilität und Nutzungseffizienz

☒ sehr gut ☐ gut ☐ mässig ☐ schlecht

Ästhetik und minimalistischer Aufbau

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben

☐ sehr gut ☒ gut ☐ mässig ☐ schlecht

Insights

- Benutzer muss wissen, wie lange er noch warten muss, bis er an der Reihe ist

Redesign Entscheide

- Warteliste einfügen

13.1.3 Prototyp Finale Version

Figure 19: Homescreen

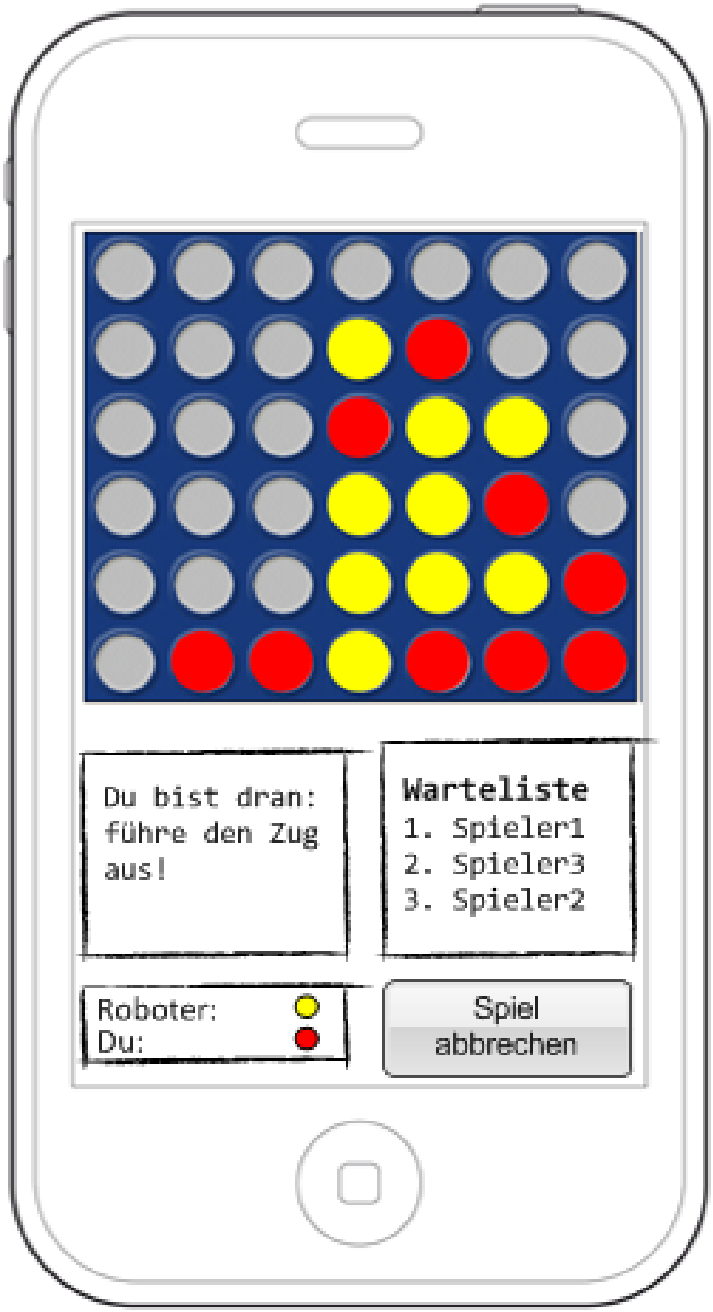


Figure 20: iPhone Prototyp Finale Version (Portrait)

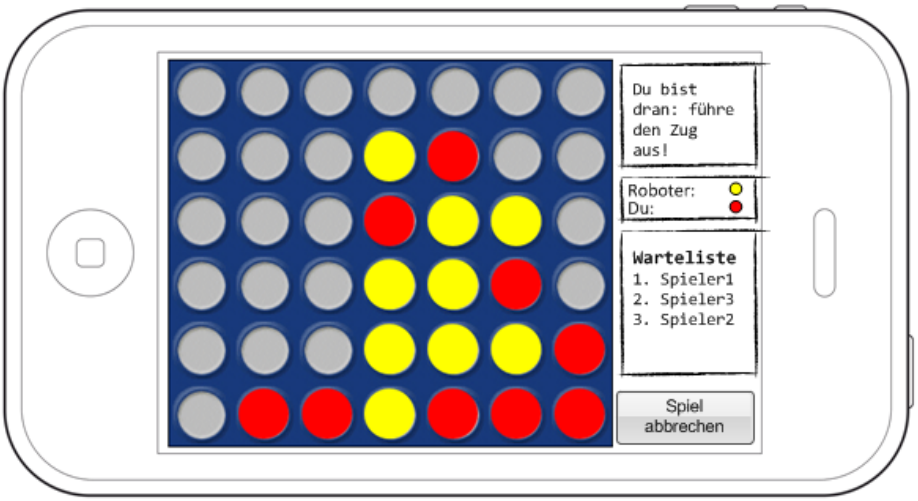


Figure 21: iPhone Prototyp Finale Version (Landscape)

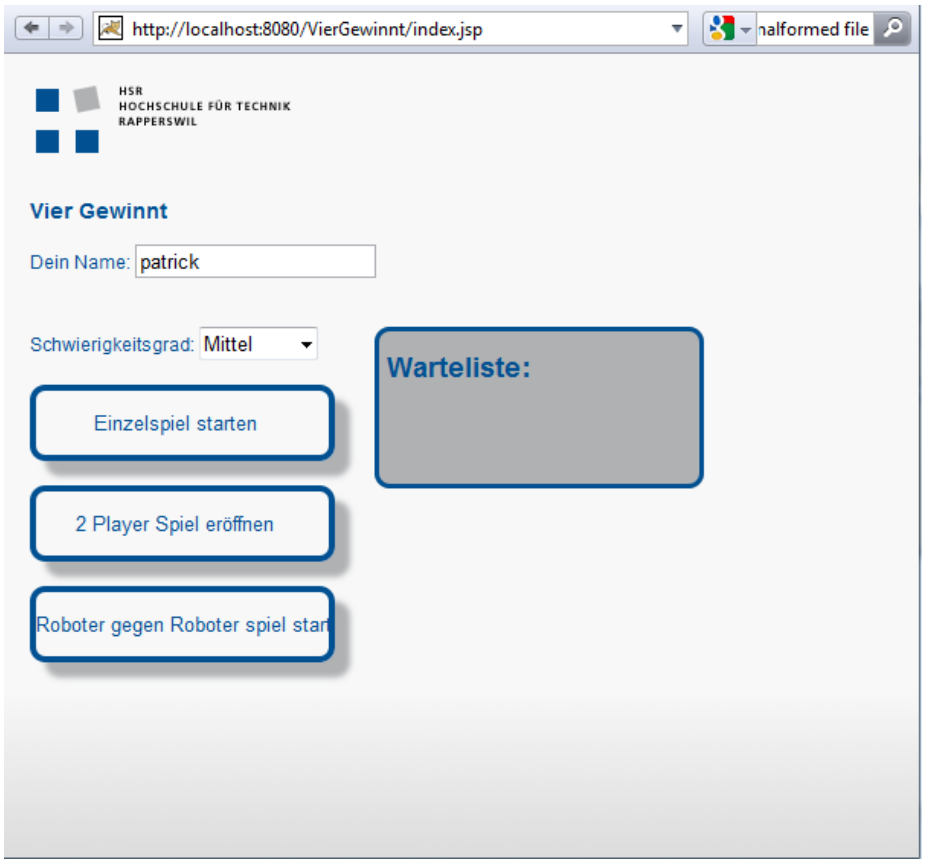


Figure 22: Homescreen im Browser

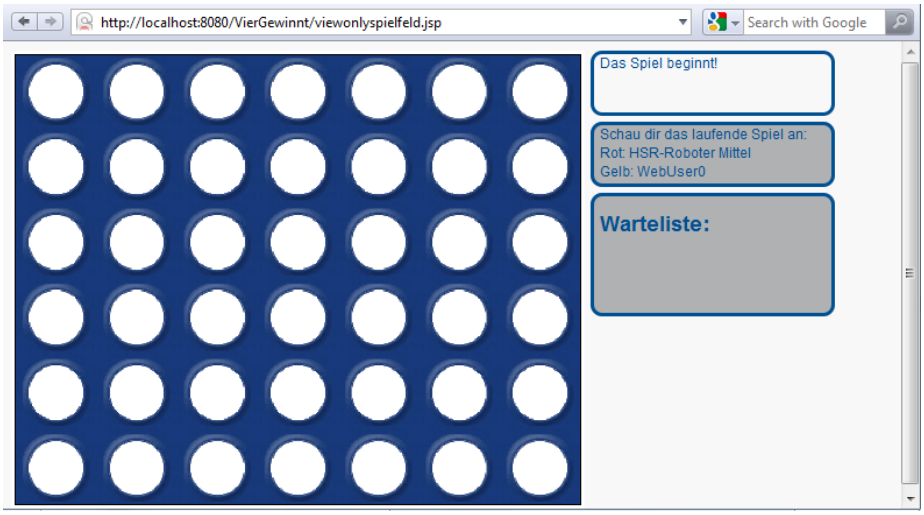


Figure 23: Spielfeld im Browser

Beim finalen Prototyp haben wir keine heuristische Evaluation gemacht, sondern uns direkt mit den GUI-Spezialisten Silvan Gehrig und Kevin Gaunt in Verbindung gesetzt, damit sie uns noch Input geben können. Die Redesign Entscheide sind im nächsten Kapitel (13.1.3.1) ersichtlich.

13.1.3.1 Redesign Entscheide

- Buttons und Informationboxen müssen grössere Unterschiede aufweisen
- Es sollten weniger Information pro Webseite enthalten sein, damit auch iPhone-User ohne Probleme die Webseite bedienen können
- Für iPad Schrift grösser machen
- Texte müssen prägnanter sein
- Letzter gespielter Stein markieren

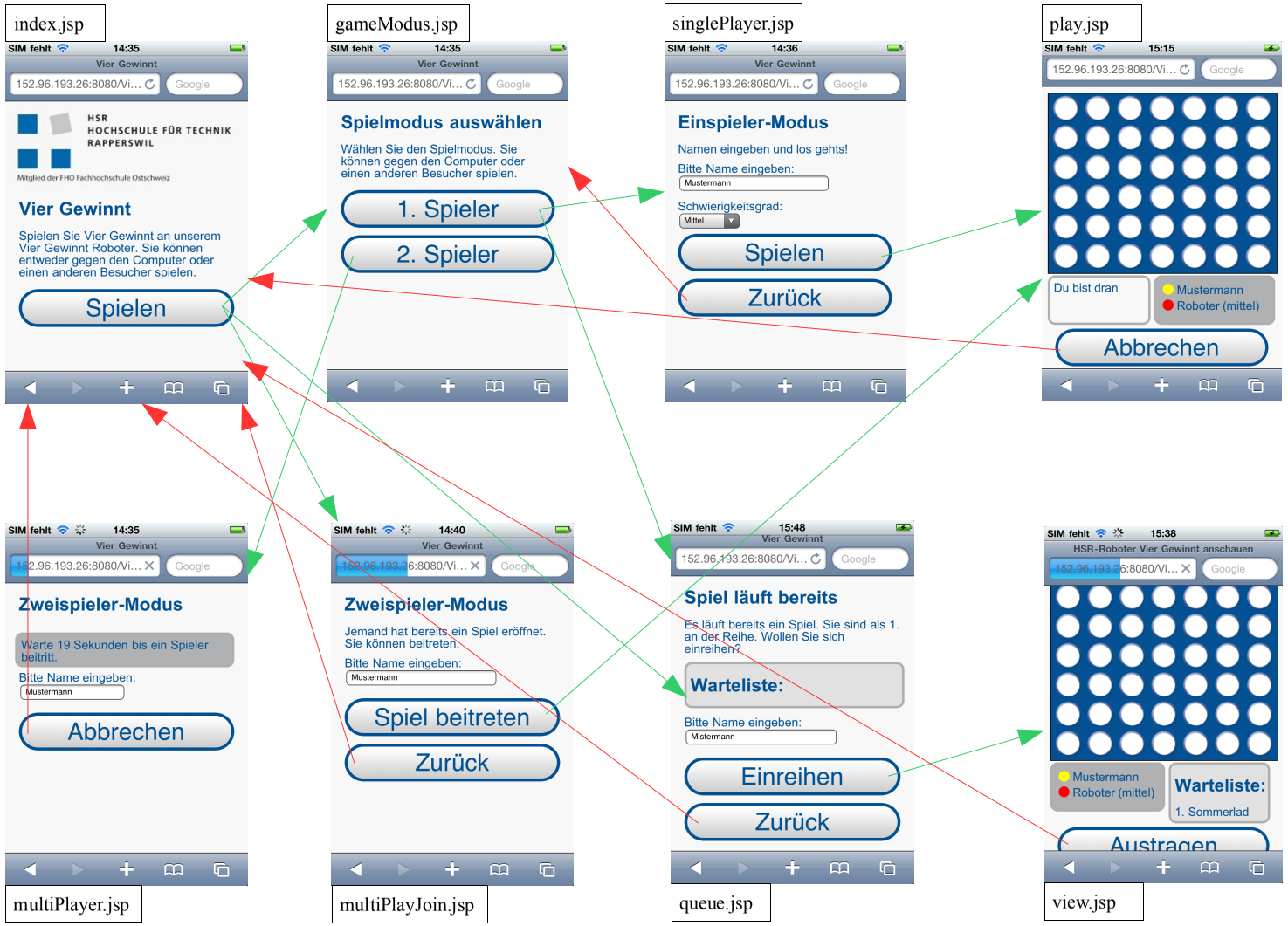
13.2 Navigation Map

Die Navigation Map zeigt wie der User von einem Screen zum anderen und wieder zurück kommt. Ausgangspunkt für weitere Navigationen ist die Seite index.jsp. Hier sind die folgenden vier "Sunny-Day"-Abläufe beschrieben:

- Spieler öffnet Einzelspiel
- Spieler öffnet 2. Spieler-Spiel
- Spieler tritt 2. Spieler-Spiel bei

- Es läuft ein Spiel und Spieler trägt sich in Warteschlange ein

Falls der Roboter nicht erreichbar ist, wir eine Fehler-Seite aufgerufen, welche im Stil von “index.jsp” ist.


index.jsp
gameModus.jsp
singlePlayer.jsp
play.jsp
multiPlayer.jsp
multiPlayJoin.jsp
queue.jsp
view.jsp
Vier Gewinnt
HSR HOCHSCHULE FÜR TECHNIK RAPPERSWIL
Mitglied der FHO Fachhochschule Ostschweiz
Spielen Sie Vier Gewinnt an unserem Vier Gewinnt Roboter. Sie können entweder gegen den Computer oder einen anderen Besucher spielen.
Spielen
Spielmodus auswählen
Wählen Sie den Spielmodus. Sie können gegen den Computer oder einen anderen Besucher spielen.
1. Spieler
2. Spieler
Einspieler-Modus
Namen eingeben und los gehts!
Bitte Name eingeben:
Mustermann
Schwierigkeitsgrad:
Mittel
Zurück
Du bist dran
Mustermann
Roboter (mittel)
Abbrechen
Zweispieler-Modus
Warte 19 Sekunden bis ein Spieler beitritt.
Spiel beitreten
Jemand hat bereits ein Spiel eröffnet. Sie können beitreten:
Spiel läuft bereits
Es läuft bereits ein Spiel. Sie sind als 1. an der Reihe. Wollen Sie sich einreihen?
Warteliste:
Einreihen
1. Sommerlad
Austragen

13.2.1 Spieler öffnet Einzelspiel

Falls der User alleine spielen will, wählt er auf der Seite gameModus.jsp "1. Spieler". Anschliessend wird die nächste Seite angezeigt, auf welchem der User seinen Namen und die Schwierigkeitsstufe eingegeben kann. Danach kann das Spiel beginnen. Für den Fall, dass bereits ein Spiel läuft siehe Kapitel 13.2.4.

13.2.2 Spieler öffnet 2. Spieler-Spiel

Wenn der User ein Spiel gegen einen anderen Infotag-Besucher spielen will, wählt er "2. Spieler". Danach gelangt der User auf die Seite "multiPlayer.jsp". Hier startet ein Counter, welche 40 Sekunden läuft, um auf einen zweiten Spieler zu warten (siehe Kapitel 13.2.3). Falls in dieser Zeit ein Spieler beitritt, wird der User direkt auf das Spielfeld (play.jsp) geleitet. Anderenfalls wird er auf die Startseite weitergeleitet.

13.2.3 Spieler tritt 2. Spieler-Spiel bei

Wie im Kapitel 13.2.2 ist die Voraussetzung, dass man einem Spiel beitreten kann, das bereits ein anderer Infotag-Besucher ein Spiel eröffnet hat und am Warten ist. In diesem Fall kann der beitretende Spieler seinen Namen eingebeben, beitreten und das Spiel wird gestartet.

13.2.4 Es läuft ein Spiel und Spieler trägt sich in Warteschlange ein

Falls bereits ein Spiel am Laufen ist, hat der User die Möglichkeit sich in die Warteliste einzutragen. Hier wird er, nachdem er sich in eingereiht hat auf die Seite "view.jsp" weitergeleitet, auf welche er das laufende Spiel beobachten kann. Nachdem das laufende Spiel beendet oder abgebrochen worden ist, wird beim wartenden Spieler bzw. bei den ersten zwei Spielern in der Warliste nachgefragt, ob er / sie nun spielen will / wollen. Nachdem die Aufforderung bestätigt wurde, wird ein Spiel gestartet

14 Architektur Design

14.1 Algorithmus

14.1.1 Einleitung

Für die Implementierung der Computerintelligenz des Spiels Vier Gewinnt stehen zwei Algorithmus zur Verfügung. Zum einen der Minimax-Algorithmus zum anderen Branch-and-Bound-Algorithmus.

14.1.1.1 Minimax Der Minimax-Algorithmus ist ein Algorithmus zur Ermittlung der optimalen Spielstrategie für bestimmte Spiele, bei denen zwei gegnerische Spieler abwechselnd Züge ausführen (z. B. Vier gewinnt), insbesondere für Nullsummenspiele. Nullsummenspiele beschreiben Spiele im verallgemeinerten Sinne, bei denen die Summe der Gewinne/Verluste aller Spieler zusammengenommen gleich null ist. Die Minimax-Strategie sichert bei Nullsummenspielen den höchstmöglichen Gewinn bei optimaler Spielweise des Gegners (das aus den Minimax-Strategien beider Spieler gebildete Strategie-Paar bildet ein Nash-Gleichgewicht. Nash-Gleichgewicht beschreibt in nicht-kooperativen Spielen einen Zustand eines strategischen Gleichgewichts, von dem ausgehend kein einzelner Spieler für sich einen Vorteil erzielen kann, indem er einseitig von seiner Strategie abweicht).[Rustem02]

14.1.1.1.1 Alpha-Beta-Pruning Alpha-Beta-Pruning ist eine optimierte Variante des Minimax-Suchverfahrens, also eines Algorithmus zur Bestimmung eines optimalen Zuges bei Spielen mit zwei gegnerischen Parteien. Während der Suche werden zwei Werte Alpha und Beta aktualisiert, die angeben, welches Ergebnis die Spieler bei optimaler Spielweise erzielen können. Mit Hilfe dieser Werte kann entschieden werden, welche Teile des Suchbaumes nicht untersucht werden müssen, weil sie das Ergebnis der Problemlösung nicht beeinflussen können.

Der Alpha-Beta-Algorithmus verwendet zwei weitere Parameter, die während der Suche verwendet und manipuliert werden: Alpha repräsentiert den höchsten bisher bekannten Gewinn, den der am Zug befindende Spieler auf irgendeine Art und Weise erzwingen kann – alles, was kleiner oder gleich alpha ist, ist nutzlos, da man durch den bisherigen Suchverlauf bereits eine Strategie kennt, die aus Sicht des aktuellen Spielers alpha erzielt. Alpha stellt also eine untere Schranke dar. Beta hingegen repräsentiert den höchsten bisher bekannten Verlust, den der Gegner des aktuellen Spielers akzeptieren muß – es ist also bereits eine Strategie für den Gegner bekannt, ein Ergebnis zu erzielen, das aus seiner Sicht nicht schlechter als beta ist. Beta stellt also für den aktuellen Spieler eine obere Schranke dar. Wenn die Suche einen Pfad betrachtet, der für den ziehenden Spieler einen Wert von beta oder mehr zurückgibt, ist dieser Pfad zu gut – der betreffende Spieler wird keine Chance zur Verwirklichung einer solchen Strategie bekommen. [Baier06]

14.1.1.2 Branch-and-Bound Das Branch-and-Bound Verfahren ist eine Variante des Backtrackings. Während beim Backtracking eine erschöpfende Suche durchgeführt wird, um alle Lösungsmöglichkeiten auf Optimalität zu prüfen, kann das Suchfeld durch das Branch-and- Bound Verfahren eingeschränkt werden. Teillösungen, die zu keiner optimalen Lösung führen können, werden vernachlässigt.

Ein Branch-and-Bound Algorithmus versucht ein gegebenes Problem bei gleichzeitiger Minimierung einer Ziel- oder Kostenfunktion zu lösen. Falls ein Problem nicht unmittelbar lösbar ist, wird das Problem in ein oder mehrere Teilprobleme zerlegt. Diese werden durch Einschränkung des übergeordneten Problems gebildet. Die Teilprobleme werden so lange zerlegt, bis sie eine zulässige Lösung darstellen oder gesichert ist, dass sie zu keiner optimalen Lösung führen. Dieser Vorgang wird als *Branching* bezeichnet.

Um zu erkennen, dass ein Teilproblem nicht zu einer optimalen Lösung führen kann, wird zu jedem Teilproblem eine untere Schranke berechnet. Diese gibt an, wie hoch die Kosten für jede mögliche Lösung des Teilproblems mindestens sind. Ist bereits eine Lösung gefunden worden, brauchen nur noch die Teilprobleme untersucht werden, deren untere Schranken kleiner sind als die Kosten x der besten bereits gefundenen Lösung. Da somit einige der Teilprobleme nicht weiter untersucht werden, wird diese Technik auch als *Bounding* (Begrenzen) bezeichnet.
Der Zerlegungsprozess des Problems lässt sich durch einen Suchbaum darstellen. Die Knoten des Baumes entsprechen den Teilproblemen und die Kanten beschreiben die Zerlegungsschritte von der Wurzel zu den Blättern. Die Wurzel repräsentiert das Gesamtproblem und die Blätter entsprechen gelösten oder nicht weiter untersuchten Teilproblemen.

Während der Zerlegung enthält der Suchbaum meist mehrere noch nicht untersuchte Teilprobleme. Das nächste zu untersuchende Teilproblem muss durch eine Suchstrategie bestimmt werden. Diese legt die Reihenfolge fest, in der der Baum durchlaufen wird. Mögliche Suchstrategien sind die Tiefensuche (depth-first), die Breitensuche (breadth-first) und die Bestensuche (best-first). Die vier wesentlichen Bestandteile eines Branch-and-Bound Algorithmus sind:

- Eine Regel zur Generierung von Teilproblemen,
- eine Auswahlregel für nicht untersuchte Teilprobleme,
- eine Regel zur Eliminierung uninteressanter Teilprobleme und
- eine Abbruchbedingung.

14.1.2 Branch, die Verzweigung

Der Branch-Schritt dient dazu, das vorliegende Problem in zwei oder mehr Teilprobleme aufzuteilen, die eine Vereinfachung des ursprünglichen Problems darstellen. Durch rekursives Ausführen des Branching-Schritts für die erhaltenen Teilprobleme entsteht eine Baum-Struktur. Es gibt verschiedene Auswahlver-

fahren für die Wahl des nächsten zu bearbeitenden Knotens im Branching-Baum, die unterschiedliche Zielsetzungen haben. Im Folgenden werden zwei häufig verwendete Verfahren beschrieben:

Tiefensuche: Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches als letztes eingefügt wurde (Last In – First Out). Mit dieser Auswahlregel arbeitet sich das Verfahren im Baum möglichst schnell in die Tiefe, so dass im allgemeinen sehr schnell eine zulässige Lösung erlangt wird, über deren Qualität jedoch nichts ausgesagt werden kann.

Breitensuche: Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches als erstes in den Baum eingefügt wurde (First In – First Out). Bei Verwendung dieser Auswahlregel werden die Knoten im Baum pro Ebene abgearbeitet, bevor tiefer gelegene Knoten betrachtet werden. Eine zulässige Lösung wird in der Regel erst relativ spät erhalten, hat aber tendenziell einen guten Lösungswert. Diese Strategie führt auch tendenziell früh zu brauchbaren unteren Schranken.

Die Verfahren sind kombinierbar. Eine erste Lösung lässt sich zum Beispiel mit einer Tiefensuche bestimmen, um dann bei einer anschließenden Bestensuche bereits eine globale obere bzw. untere Schranke zu haben.

14.1.3 Bound, die Schranke

Der Bound-Schritt hat die Aufgabe, bestimmte Zweige des Baumes "abzuschneiden", d. h. in der weiteren Berechnung nicht mehr zu betrachten, um so den Rechenaufwand zu begrenzen. Dies erreicht der Algorithmus durch Berechnung und Vergleich zweier Schranken. Obere Schranken ergeben sich aus jeder zulässigen Lösung. Dafür kann gegebenenfalls zuvor eine Heuristik benutzt werden.

14.2 Komplexität / Laufzeit

14.2.1 Minimax

Komplexität O(b^m) mit:

- b: Anzahl der erlaubten Spielzüge pro Knoten
- m: Anzahl der Züge die vorausberechnet werden

Zum Beispiel: 7^42 Endstellungen bei vollständigem MiniMax -> ca. 3*10^35 Möglichkeiten. Laufzeit expotentiell steigend mit der Zuganzahl, aber sehr stark von der Bewertungsfunktion abhängig -> schnellere Bewertungsfunktion, große Zeitersparnis.

14.2.2 Branch-and-Bound

Komplexität O(e^n)

- Branch-and-bound Verfahren sind im schlechtesten Falle exponentiell. Das heisst wenn kein Zweig eines Baumes “abgeschnitten” werden kann.

14.2.3 Vergleich

Kritieren	Gewichtung	Minimax	Branch-and-Bound
Laufzeit	3	2	5
Implementation	3	4	3
Schwierigkeitsstufen	4	5	2
Total (Gewichtung * Wert)		38	32

Gewichtung: 1: nicht wichtig, 5: sehr wichtig

14.2.3.1 Fazit Branch-and-Bound ist für allgemeine Aufgaben, wie Optimierung von Prozessen, ähnlich dem A*-Algorithmus, gedacht. Wohingegen der Minimax-Algorithmus insbesondere für Nullsummenspiele entwickelt wurde.

Der Branch-and-Bound Algorithmus hat die bessere Laufzeit. Allerdings ist die bei Minimax leichter die Schwierigkeitsstufen einzustellen und dadurch die ganze Implementation einfacher, da nicht ein Parameter eingefügt werden muss, welche die Schwierigkeitsstufen repräsentiert. Die leicht schlechtere Laufzeit kann in Kauf genommen werden. Falls die Laufzeit doch zu schlecht sein sollte, kann der Minimax-Algrithums durch das Anwenden von Alpha-Beta-Pruning optimiert werden.

14.2.3.2 Entscheid Wie im Vergleich und im Fazit ersichtlich, bietet Minimax mehrere Vorteile gegenüber Branch-and-Bound. Darum entscheiden wir uns den Minimax-Algorithmus zu implementieren.

Pseudocode Minimax

```
Hauptprogramm (Auszug):
  var doNext : number
  dummy := maxWert ( gewünschte suchTiefe )
  Zug doNext ausführen

function maxWert ( restTiefe ) returns number
var ermittelt, zugWert : number
begin
  ermittelt := - unendlich
  für alle möglichen Züge begin
    Zug simulieren
    if restTiefe <= 1 or keineFolgezügeMehrMöglich
    then zugWert := bewertungsFunktion
    else zugWert := minWert ( restTiefe - 1 )
    Zug-Simulation zurücksetzen
    if zugWert > ermittelt then begin
      ermittelt := zugWert
      if restTiefe = gewünschte suchTiefe
        doNext := nummer des Zuges
```

```
    end
  end
  return ermittelt
end maxWert

function minWert ( restTiefe ) returns number
var ermittelt, zugWert : number
begin
  ermittelt := + unendlich
  für alle möglichen Züge begin
     Zug simulieren
    if restTiefe <= 1 or keineFolgezügeMehrMöglich
    then zugWert := bewertungsFunktion
    else zugWert := maxWert ( restTiefe - 1 )
    Zug-Simulation zurücksetzen
    if zugWert < ermittelt then ermittelt := zugWert
  end
  return ermittelt
end minWert
```

[url01]

14.3 Schnittstellenbeschreibung

14.3.1 Daten vom Roboter zum Server

Die Daten in folgender Tabelle sind Antworten, welche der Roboter dem Server als Meldungen senden können muss.

Name	Beschreibung	char Wert welcher der Roboter dem Server sendet:
Ready	Bereit für den nächsten Zug	1
Busy	Abarbeiten des aktuellen Zuges	2
Error	Es ist ein Fehler aufgetreten	3
Received	Die aktuelle Aufgabe wurde erhalten	4

Tabelle 11: Befehle Roboter -> Server

14.3.2 Daten vom Server zum Roboter

Die Daten in folgender Tabelle sind Befehle oder Zustandsabfragen, welche der Server dem Roboter senden können muss. Ebenfalls aufgeführt sind die möglichen Antworten des Roboters.

Name	Beschreibung	char Wert welcher der Server dem Roboter sendet:	Antwort des Roboters
Status	Den aktuellen Status des Roboters abfragen	A	Ready(char 1), Busy(char 2), Error(char 3)
Initialize	Den Roboter neu inizialisieren. Allenfalls eine Achse neu einstellen. Wird beim starten des Servers oder nach einem Error des Roboters aufgerufen	B	Received(char 4)
New	Neues Spiel starten. Bei diesem Befehl muss das Spielfeld geleert werden.	C	Received(char 4)
DRAW	Das spiel endet unentschieden	D	Received(char 4)
RW	Rot gewinnt	E	Received(char 4)
YW	Gelb gewinnt	F	Received(char 4)
R1-R7	Roter Stein in Feld 1-7 einfügen	G, H, I, J, K, L, M	Received(char 4)
Y1-Y7	Gelber Stein in Feld 1-7 einfügen	N, O, P, Q, R, S, T	Received(char 4)
WP	Gewinnpositionen (für LED anzeige) Das Spielfeld ist dafür von unten Links (char U) bis oben rechts (char 0) Zeile für Zeile durchnummeriert.	U, V, W, X, Y, Z, a-z, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0	Received(char 4)

Tabelle 12: Befehle Server -> Roboter

14.3.3 Kommunikation

Für die Kommunikation mit dem Roboter wurden verschiedene konzeptionelle Modelle angeschaut, welche nachfolgend kurz beschrieben werden.

14.3.3.1 Synchrone Kommunikation (nur vom Server aus): Der Server öffnet eine Verbindung zum Roboter und sendet diesem einen Befehl. Der Roboter führt den Befehl aus und sendet nach der Abarbeitung ein "Success" oder "Error" zurück. Wenn der Roboter lange nicht reagiert wird der Befehl erneut gesendet. Während der ganzen Zeit des Befehls wartet der Server.

Schema:

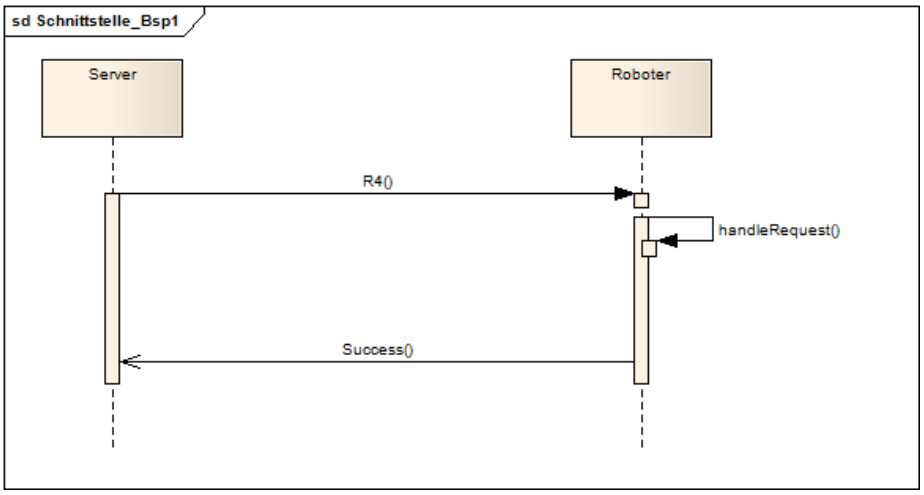


Abbildung 24: Synchrone Kommunikation

Vorteil:

- Es werden relativ wenig Daten ausgetauscht.

Nachteil:

- Bei einem Fehler muss lange gewartet werden.

14.3.3.2 Asynchrone Kommunikation: Der Server öffnet eine Verbindung zum Roboter und sendet diesem einen Befehl. Dieser führt den Befehl aus und sendet nach der Abarbeitung ein "Success" oder "Error" zurück. Nachdem der Server den Befehl gesendet hat, kümmert er sich nicht mehr darum sondern geht anderen Tätigkeiten nach. Sobald der Roboter fertig ist mit der Aufgabe meldet er sich wieder beim Server mit einer "Success" oder "Error" Meldung.

Schema:

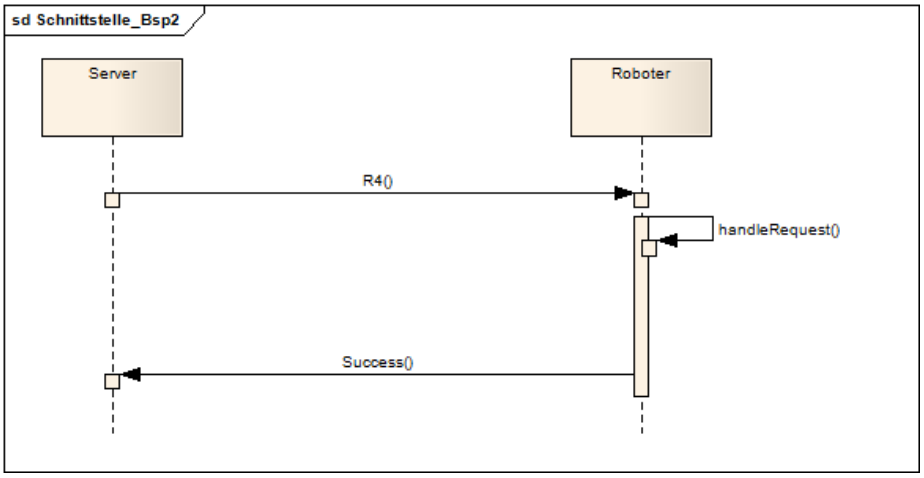


Abbildung 25: Asynchrone Kommunikation

Vorteil:

- Der Server ist nicht während des ganzen Aufruf blockiert, sondern kann anderen Aufgaben nachgehen.

Nachteil:

- Wenn der Auftrag nicht richtig übertragen wurde, kann es sein, dass der Roboter sich nicht mehr beim Server meldet und der Server dies gar nicht merkt, da er nicht auf eine Antwort wartet.

14.3.3.3 Synchrone Kommunikation (von Server und Roboter) Der Server öffnet eine Verbindung zum Roboter und sendet diesem einen Befehl. Der Roboter sendet umgehend ein "Received" zurück und erledigt den Auftrag. Der Server kann nun an etwas anderem arbeiten. Wenn der Roboter mit seinem Auftrag fertig ist, sendet er dem Server eine "Success" oder eine "Error" Meldung zurück. Diese wird vom Server wiederum direkt mit einem "Received" beantwortet.

Schema:

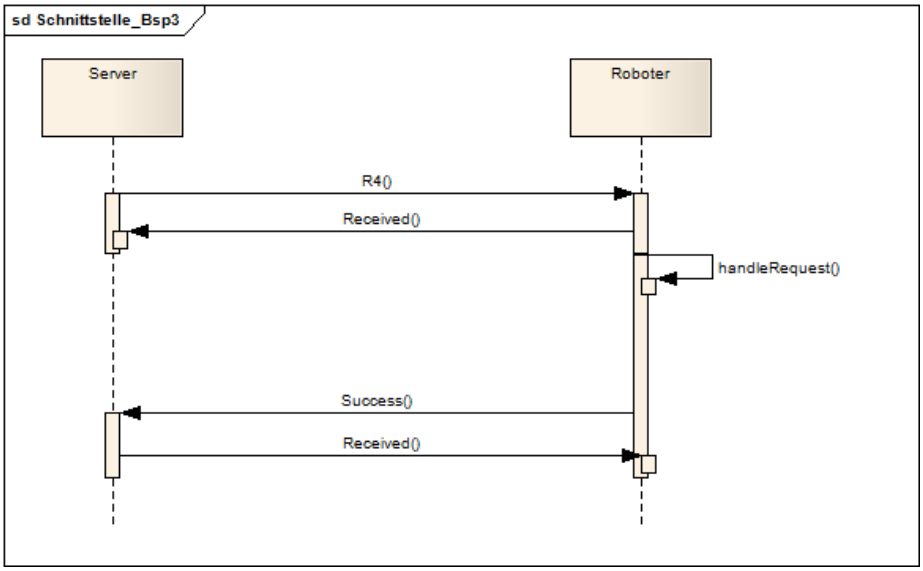


Abbildung 26: Synchrone Kommunikation (von Server und Roboter)

Vorteil:

- Die Systeme geben sich immer sofort eine Antwort.
- Das nicht erhalten eines Befehls wird schnell erkannt und der Befehl kann nochmals gesendet werden.
- Ein Abfragen des Systemstatus der anderen Seite ist einfach möglich.

Nachteil:

- Die Implementation auf der Seite des Roboters ist ein wenig aufwändiger.
- Es könnte zu Deadlocks kommen wenn beide auf einander warten.

14.3.3.4 Synchrone Kommunikation mit Server Polling Die in Kapitel 14.3.3.3 beschriebene Synchrone Kommunikation ist auf der Roboterseite relativ aufwändig und das öffnen der Verbindung von beiden Seiten kann zu Problemen führen. Eine weitere Möglichkeit ist, dass die Kommunikation immer nur vom Server gestartet wird. Der Roboter sendet auf jeden Befehl des Servers direkt ein "Received". Der Server sendet nach einem Befehl in regelmässigen Abständen eine Statusabfrage, auf die der Roboter wiederum sofort mit einem Ready, Busy oder Error antwortet.

Schema:

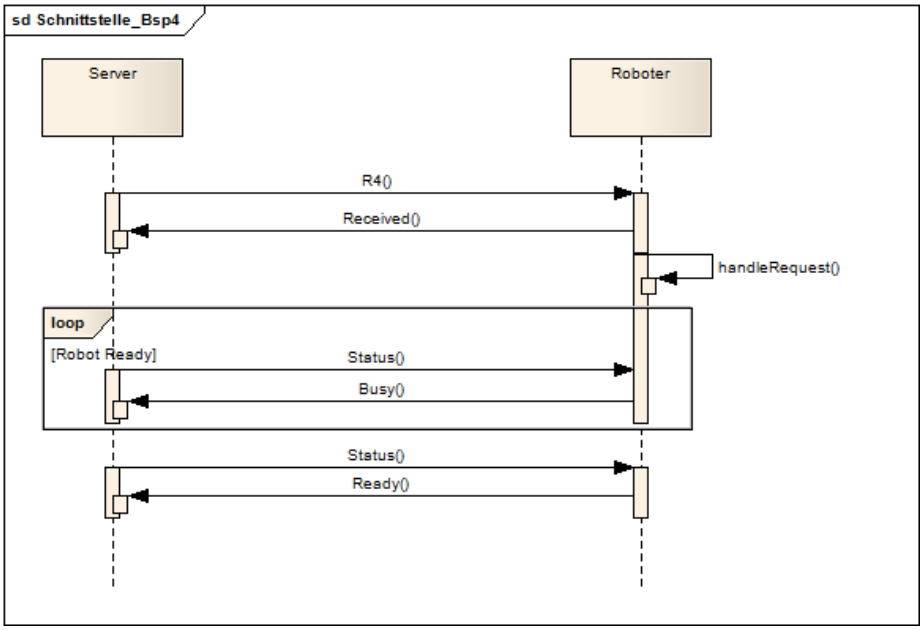


Abbildung 27: Synchrone Kommunikation mit Server Polling

Vorteil:

- Der Roboter gibt immer sofort eine Antwort.
- Das nicht erhalten eines Befehls wird schnell erkannt und der Befehl kann schnell nochmals gesendet werden.
- Die Programmierung auf der Roboterseite ist relativ einfach.

Nachteil:

- Durch das stetige Abfragen des Roboterstatus, gibt es relativ viel zusätzlichen Traffic.

14.3.3.5 Entscheid Nach Absprache mit Prof. Dr. Markus Stolze haben wir uns für die synchrone Kommunikation mit Server Polling entschieden (Kap.14.3.3.4). Die Entscheidung für diese Lösung fiel, da diese uns als sehr robust erscheint. Die Kommunikation wird immer vom Server aus gestartet und mit einer einseitigen Kommunikation sind Deadlocks nicht möglich.

14.3.4 Analyse Nachrichtenaustausch

In diesem Kapitel wird für die gewählte Kommunikation der Nachrichtenaustausch beschrieben. Als erstes in einem fehlerfreien Ablauf und danach in verschiedenen fehlerhaften Abläufen.

14.3.4.1 Fehlerfreie Kommunikation In der Abbildung 28 sieht man die Sequenz einer fehlerfreien Kommunikation. Der Server sendet zuerst dem Roboter den Befehl "R4", also einen roten Stein in die vierte Spalte zu legen. Der Roboter sendet darauf eine "Received" Meldung an den Server um mitzuteilen, dass er den Befehl erhalten hat. Danach führt er den Befehl aus. Der Server fragt in regelmässigen Abständen nach, ob der Roboter den Befehl schon ausgeführt hat. Der Roboter antwortet darauf mit einem "Busy" wenn er noch am Arbeiten ist oder mit einem "Ready" wenn er nichts mehr zu tun hat. Sobald der Server ein "Ready" erhält stoppt er seinen Statusabfrage-Loop.

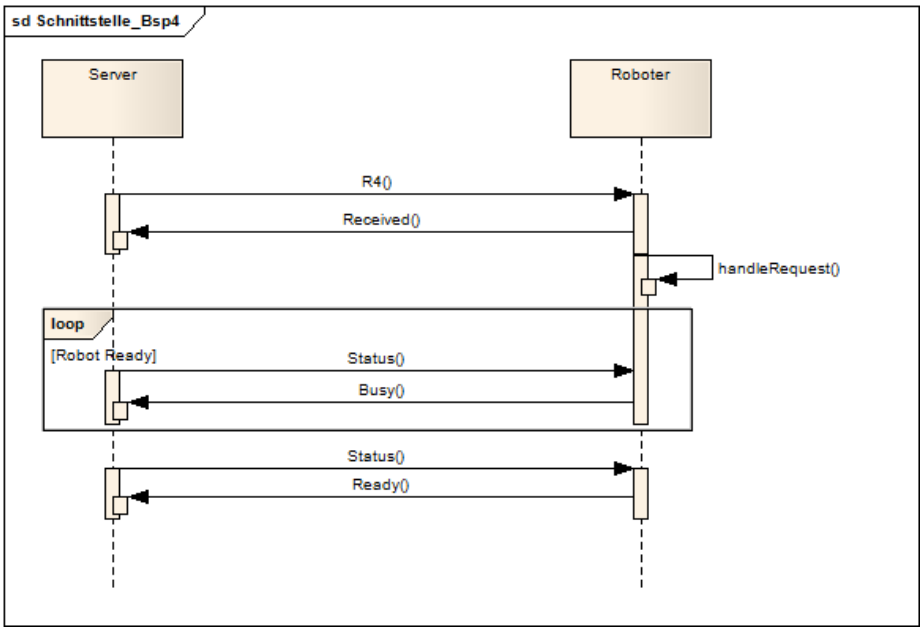


Abbildung 28: Fehlerfreie Kommunikation

14.3.4.2 Fehlerhafte Kommunikation In der fehlerhaften Kommunikation werden genau die gleichen Nachrichten ausgetauscht wie in der fehlerfreien Kommunikation (Kap. 14.3.4.1). Der Unterschied liegt darin, dass der erste Befehl nicht beim Roboter ankommt. Da der Server darauf kein "Received" vom Roboter erhält, sendet er die anfrage nochmals.

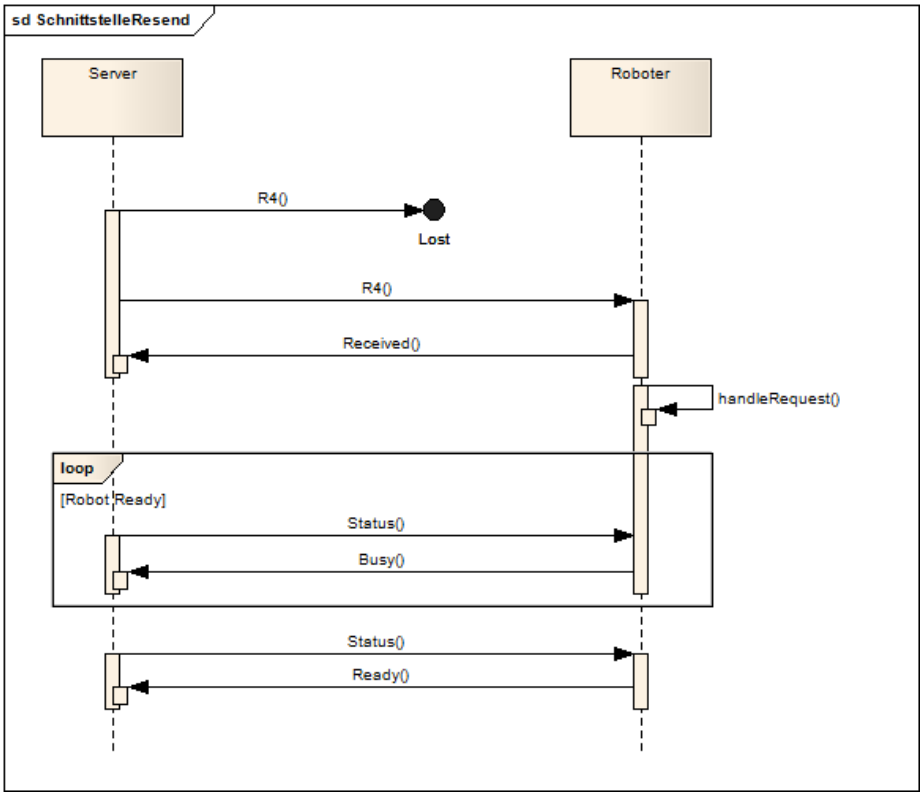


Abbildung 29: Fehlerbehaftete Kommunikation

14.3.4.3 Roboter Timeout Wenn der Server zu lange immer nur ein "Busy" zurück erhält, muss er davon ausgehen, das beim Roboter ein Fehler vorliegt. Deshalb gibt es ein Timeout nachdem der Server die Statusabfrage abbricht und versucht den Roboter neu zu starten.

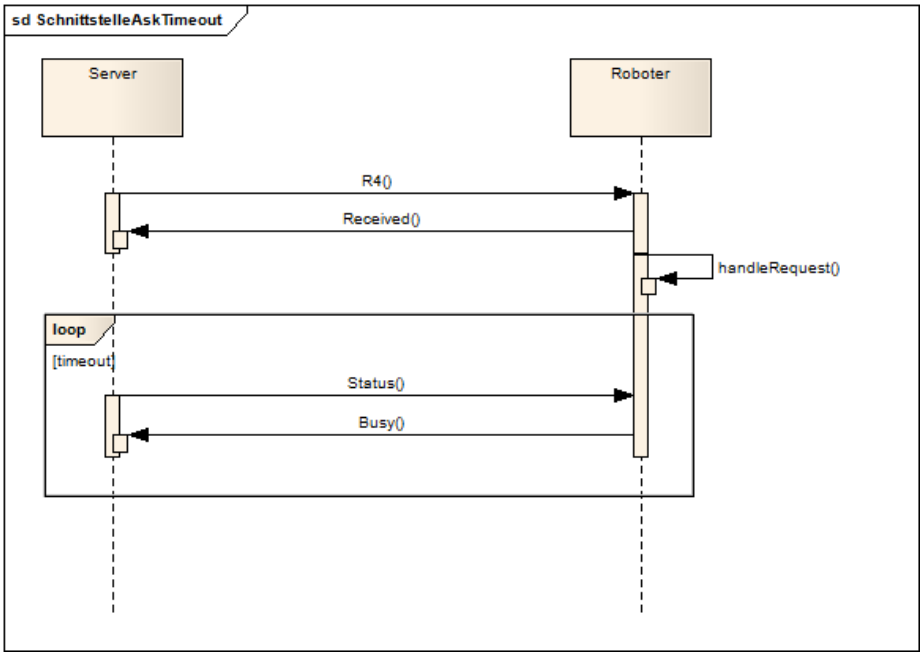


Abbildung 30: Statusabfrage auf Roboter

14.3.5 Fehler beim Roboter

Wenn wirklich ein Fehler beim Roboter vorliegt, gibt er dies bei der Statusabfrage mit einem "Error" bekannt. Je nach Fehler ist es denkbar, dass verschiedene "Error"-Nachrichten möglich sind. Der Server muss dann versuchen den Roboter neu zu starten oder er muss dem Benutzer eine Fehlermeldung anzeigen.

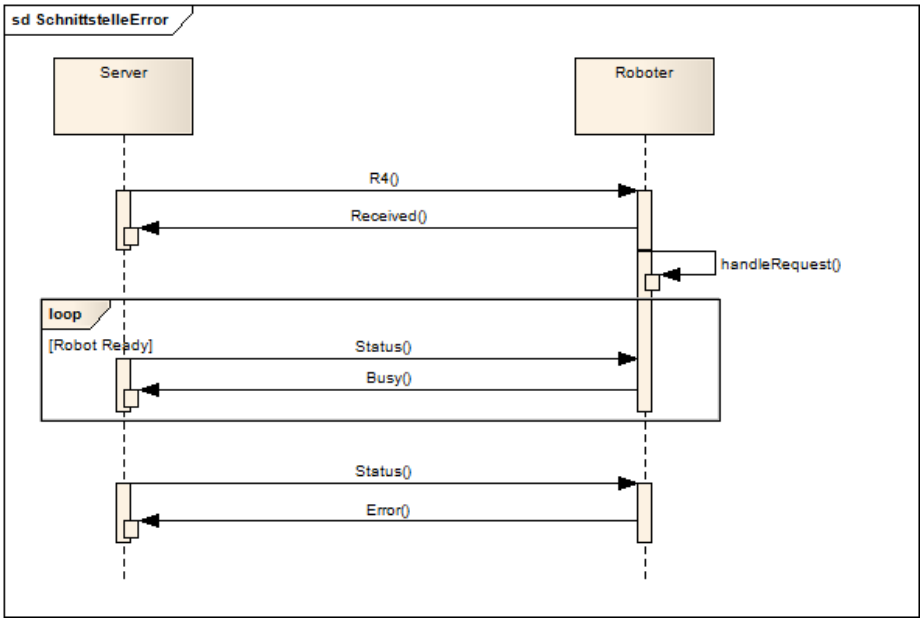


Abbildung 31: Roboter gibt Fehlercode zurück

14.3.6 Zustandsdiagramme

In den Zustandsdiagrammen wird angezeigt, in was für Zuständen der Server und der Roboter sich befinden können.

14.3.6.1 Server Bei diesem Zustandsdiagramm handelt es sich um den Zustand des Servers aus sicht des Roboters. Wenn kein Fehler vorliegt, ist es ein ständiger Wechsel zwischen warten bis der Roboter einen Befehl ausgeführt hat und warten bis ein Webuser oder die KI einen Zug durchgeführt hat.

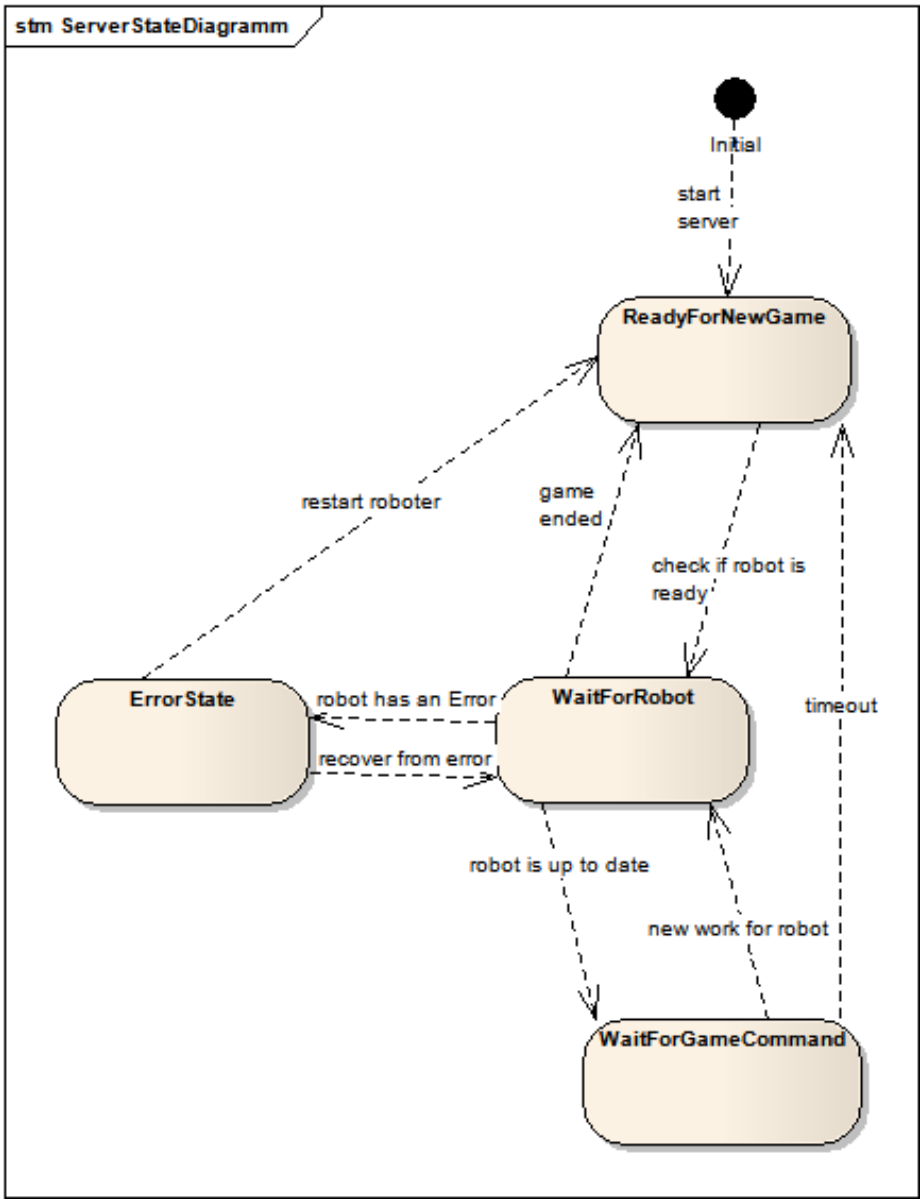


Abbildung 32: Server State Diagramm

14.3.6.2 Roboter In der folgenden Abbildung wird die verschiedenen Zustände des Roboters dargestellt.

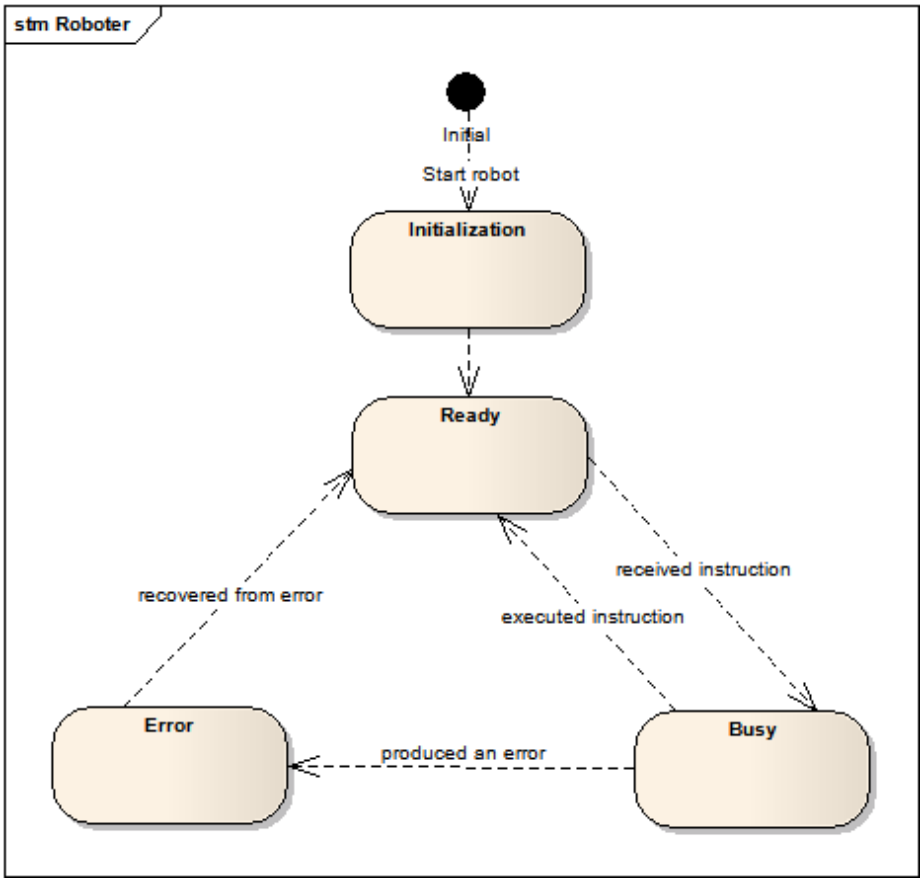


Abbildung 33: Roboter State Diagramm

14.3.7 OPC Kommunikation

Diese Möglichkeit wird voraussichtlich doch nicht realisiert. Der Vollständigkeitshalber wird der Abschnitt nicht gelöscht.

Eine mögliche Lösung für die Kommunikation zwischen Server und Roboter ist die standartisierte Software-Schnittstelle OPC (OLE for Process Control). OPC ermöglicht den Datenaustausch zwischen Anwendungen unterschiedlichster Hersteller in der Automatisierungstechnik. Dazu braucht es einen OPC Client bei der SPS und einen OPC Server auf dem Server. Unsere Applikation muss ihrerseits ebenfalls über einen OPC Client verfügen. Der OPC Client sendet Daten lokal dem OCP Server, welcher diese schliesslich über das Netzwerk dem OPC Client auf der SPS sendet. Das Ganze ist in der nachfolgenden Abbildung visualisiert.

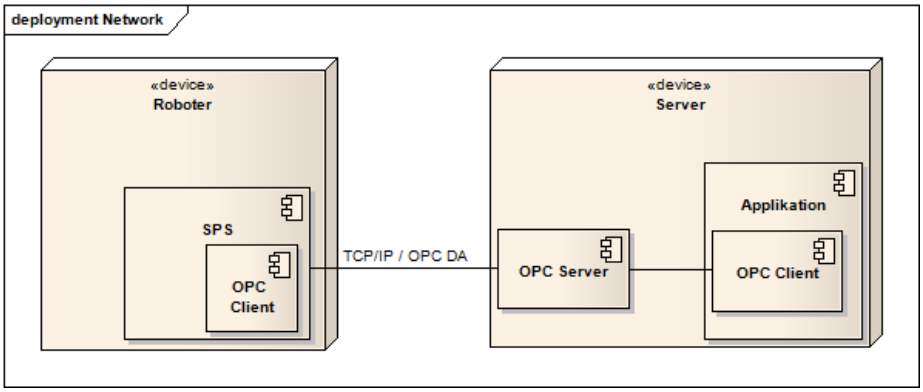


Abbildung 34: OPC Kommunikation Server <-> Roboter

14.3.8 USB-Serial Kommunikation

Auf der Hardware Ebene wird für die Verbindung zwischen Roboter und Server ein USB-Serial Converter eingesetzt. Für das Produkt gibt es Windows Treiber, damit der USB-Stecker als COM-Anschluss angesteuert werden kann. In Java gibt es von RXTX[url02] eine Open Source Java Library mit der es möglich ist, über serielle Schnittstellen zu kommunizieren. Auf der Roboter Seite muss das serielle Signal noch von 5V auf 24V transformiert werden, danach kann es als Input für die SPS genutzt werden.

14.4 UserInterface-Technologie

14.4.0.1 HTML5 Wir haben uns im Vorfeld für HTML5 entschieden, weil HTML5 eine Technologie ist, welche in naher Zukunft die bestehende Sprache im Web, HTML, ablösen wird. Darum werden wir unsere Arbeit mit der neuen, zukunftorientierer Websprache HTML5 realisieren. HTML5 bietet zahlreiche Möglichkeiten, die in HTML nur mittels proprietären Sprachen wie Flash oder Plugins (z. B. Video anschauen) umgesetzt werden konnte. HTML5 ist die Hauptspezifikation, in der die wichtigsten Grundlagen von HTML5 enthalten sind. [Kroener10]

HTML5 bietet folgende Elemente, welche wir in unserer Arbeit verwenden können:

- *audio, video*
  Audio- und Videodateien sollen zukünftig mit diesen Elementen eingebunden werden. HTML5 stellt zu diesen Elementen einige Schnittstellen bereit, beispielsweise um die Wiedergabe steuern zu können.

- *canvas*
  Das canvas-Element erzeugt eine Zeichenoberfläche, auf der mit Hilfe von DOM-Methoden gezeichnet werden kann. Die Spezifikation sieht nur eine

zweidimensionale Zeichenoberfläche vor, geht aber davon aus, dass in Zukunft auch dreidimensionale Kontexte eingeführt werden könnten.

- *progress*
  Das progress-Element gibt den Fortschritt einer bestimmten Aufgabe wieder. Das Element wird ebenfalls mit dem DOM gesteuert. Anwendungsbeispiel wäre z. B. der Fortschritt bei einer mehrseitigen Umfrage.

14.4.0.1.1 Die Standarddarstellung HTML5 versucht, die Erwartung von Autoren an die Standarddarstellung der Elemente wiederzugeben. Für alle Elemente und deren Attribute gibt es daher eine „erwartete Darstellung“, die durch CSS-Eigenschaften definiert wird. HTML5 unterscheidet dabei zwischen Darstellungseigenschaften, die bei der standardkonformen, und der kompatibilitätsorientierten Verarbeitung von Webseiten angewendet werden sollen.

14.4.0.1.2 Der Browserkontext HTML5 führt das Konzept eines Browserkontextes ein, in jedem Browserkontext wird ein Dokument geladen bzw. weitere Browserkontexte (im Fall von Frames) erzeugt. Zu den Bestandteilen eines Browserkontextes gehören großteils JavaScript-Objekte, die zuvor keinem Standard angehörten, z. B. das History-Objekt, in dem die Abfolge der besuchten Webseiten gespeichert wird. Dadurch wird versucht, das Verhalten der Browser zu vereinheitlichen und einer gemeinsamen Definition zu unterwerfen.

14.5 Logische Architektur

14.5.1 Packages

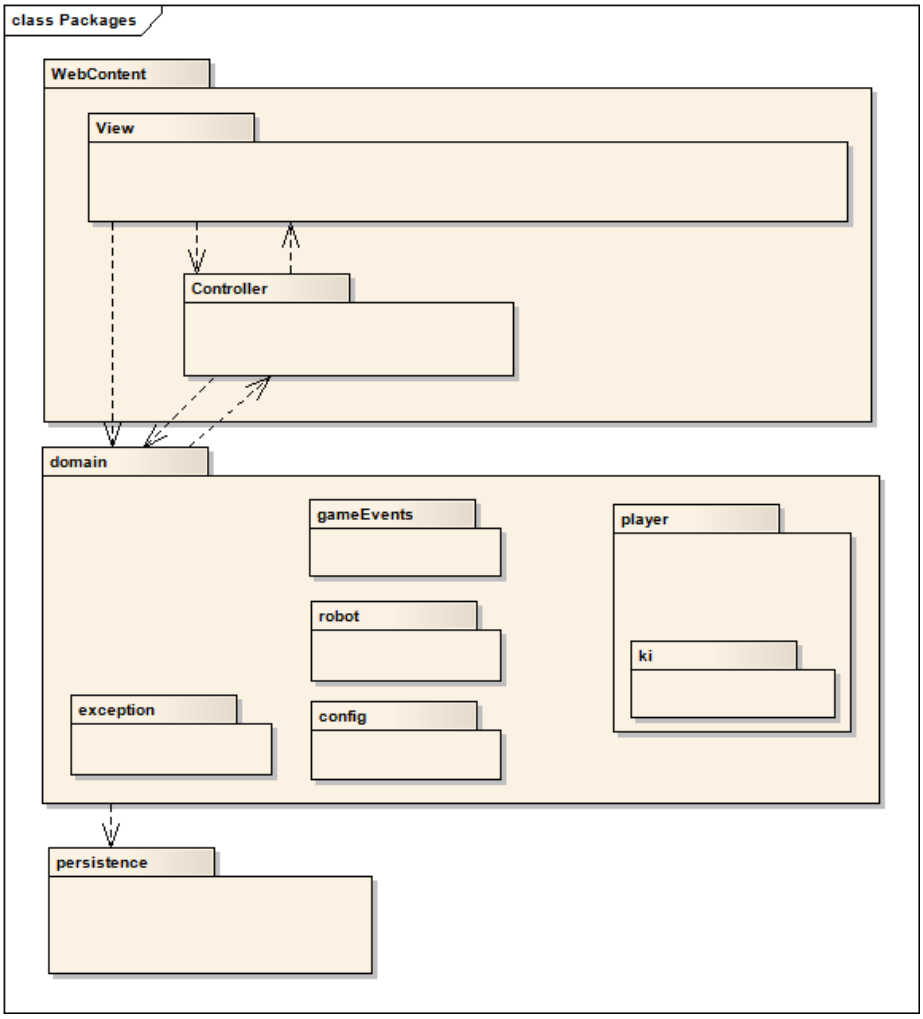


Abbildung 35: Packages

Die Applikation ist in drei logische Schichten unterteilt. Im WebContent befindet sich das GUI sowie Elemente für die Kommunikation mit dem GUI. Die Kommunikation zwischen Domain und GUI wurde nach dem MVC Pattern implementiert. Im Domain Package sind alle, für das Spiel und die Kommunikation mit dem Roboter, relevanten Klassen abgelegt. Zur besseren Übersicht wurden Teile, welche zusammengehören, in weitere Subpackages unterteilt. Im Persistence Layer ist der Zugriff auf die Datenbank implementiert. Nachfolgend

werden die verschiedenen Packages genauer beschrieben.

14.5.1.1 Package WebContent Das Package WebContent beinhaltet alle Dateien, welche für das Front-End benötigten werden. D. h. alle .jsp-Files und Javascript-Files und die Bilder, welche auf der Homepage angezeigt werden und die vier .css-Files (landscape / portrait-Modus jeweils für iPad sowie für das iPhone). Die verschiedenen Views sind im Kapitel 13 UI Design genauer beschrieben.

14.5.1.1.1 Diagramm JS Für das handling des Spiels wurde JavaScript verwendet. Alle Methoden die für den "View" und den "Play" Modus sind im File generalfunctions.js abgelegt. Dieses File wird von den anderen beiden Files geladen. Im Play sind zusätzlich alle Methoden, die mit der Zugeingabe zu tun haben enthalten. Beim Laden der Seite wird ein Eventlistener für das Canvas Objekt gemacht und bei jedem Klick wird die onClick Methode aufgerufen. Je nachdem, ob das klicken gerade erlaubt ist, wird eine Aktion ausgeführt. Mit der Methode getColumn wird ausgerechnet in welcher Spalte ein Klick stattgefunden hat.

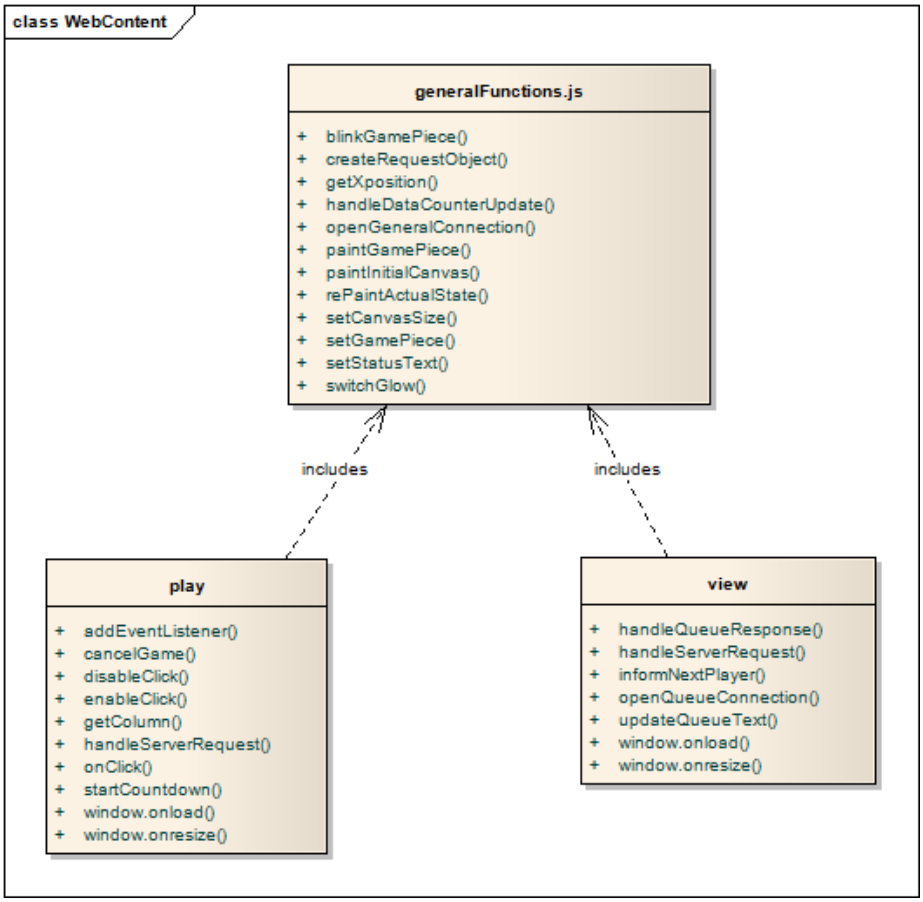


Abbildung 36: JavaScript Methoden

Im View File sind zusätzlich zu generalfunctions.js hauptsächlich Methoden für das Handling der Warteliste definiert. Die Methode informNextPlayer wird aufgerufen sobald ein Spiel fertig ist und die queueEvents die von AJAX gesendet werden ein Flag enthalten, dass der wartende User nun an der Reihe ist.

Eine genauere Beschreibung aller JS Methoden ist in den jeweiligen Files als Kommentar zu finden.

14.5.1.1.2 Spieldatenaustausch Während einem laufenden Spiel muss der User alle Änderungen vom Server immer so schnell wie möglich erhalten. Dies wurde mit einem AJAX Long Polling (Comet) realisiert. Der Browser öffnet eine AJAX Verbindung zum Server und diese bleibt so lange offen, bis auf dem Server wieder neue Daten vorhanden sind. Diese Daten werden schliesslich in JSON Notation dem Browser gesendet. Sobald dieser die Daten verarbeitet hat, öffnet er wieder eine Verbindung zum Server.

Alle GameEvents (Spielzüge, Spielabbruch, Gewinnanzeige) werden durchnummeriert. Der Browser sendet bei jeder Datenanfrage eine Zahl mit, die angibt bis zu welchem Event er die Spieldaten schon erhalten hat. Mit diesem Verfahren ist gewährleistet, dass der Browser alle Events in der richtigen Reihenfolge erhält und das bei einem Refresh der Seite alle bisherigen Events nochmals gesendet werden (da der DataCounter auf dem Browser dann wieder bei 0 anfängt).

14.5.1.1.3 Schnittstellen Die Kommunikation zwischen dem Frontend und der Domain geschieht über den Zugriff auf die Bean Klassen GameContext und WebPlayer. Je nachdem ob gerade ein Spiel läuft muss dem User eine andere Seite angezeigt werden. Dies geschieht über eine Abfrage des aktuellen Zustandes auf dem GameContext. Alle Views greifen nie schreibend auf die Domain zu. Ihre Aufrufe richten sie immer an die GameController.jsp Klasse, in welcher überprüft wird, ob die Aktion erlaubt ist und danach z. B. ein neues Spiel gestartet wird.

14.5.1.2 Package domain Das Package domain beinhaltet alle relevanten Dateien damit das Spiel läuft. Die einzelnen Teile wurden zur besseren Übersicht zusätzlich noch in Subpackages unterteilt. Damit die Diagramme übersichtlicher sind und weniger Platz brauchen, werden Getter und Setter Methoden sowie einzelne Konstanten weggelassen. Auch die Enums Colour und GameEnd, welche an verschiedenen Orten verwendet werden, sind nicht in den Diagrammen aufgeführt. Die Beschreibungen in diesem Dokument sollen eine gute Übersicht über die Architektur und das Zusammenspiel der Klassen geben. Eine viel genauere Beschreibung zu den Klassen und Methoden ist im JavaDoc zu finden, welches ebenfalls auf der abgegebenen CD zu finden ist.

14.5.1.2.1 Klassendiagramm Domain In der Domain ist der GameContext die Hauptklasse für das gesamte Programm. Da der Roboter nur ein Spiel zur gleichen Zeit laufen lassen kann, muss diese Klasse dies sicherstellen. In dieser Klasse wird auch die Queue für die wartenden Spieler verwaltet. Aus dem GameContext wird ein neues Spiel gestartet.

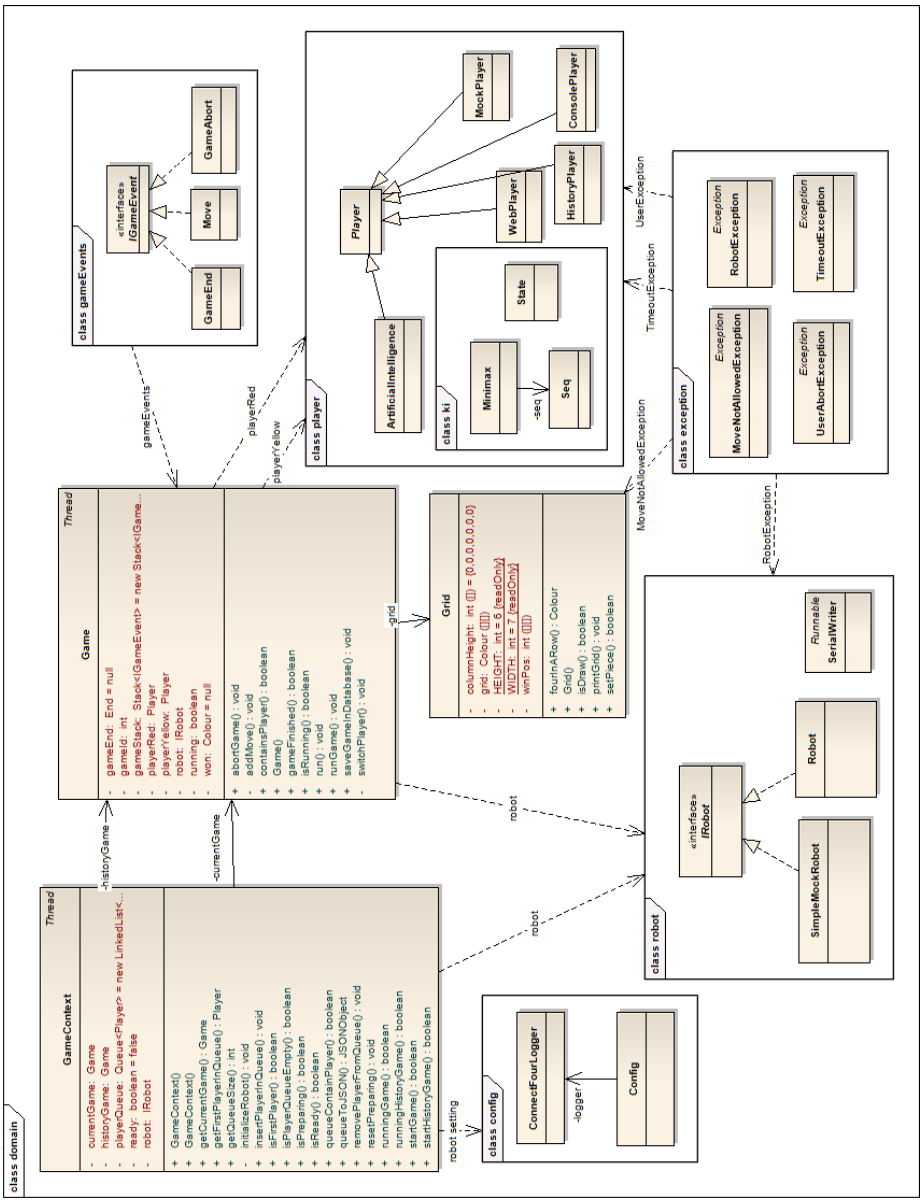


Abbildung 37: Package domain mit Subpackages

Die Klasse Game enthält jeweils ein Spiel. Hier müssen genau zwei, vom Interface Player, abgeleitete Spieler angegeben werden und eine Klasse, welche das Interface IRobot implementiert. Das Spielfeld verwaltet die Game Klasse in der Klasse Grid. Damit ein Game gestartet werden kann, ohne das eine Methode bis das Game fertig ist warten muss, wurde Game als Thread implementiert,

sodass ein Game gestartet wird und danach asynchron läuft.

14.5.1.2.2 Beans Für die Verbindung zum WebContent sind die Klassen GameContext und WebPlayer als Beans im Tomcat verfügbar. Die Klasse WebPlayer ist dabei ein Sessionbean, wird also für jeden User neu erstellt und repräsentiert den User. Die Klasse GameContext ist ein Applicationbean und ist nur einmal vorhanden. Im GameContext sieht man, ob gerade ein Spiel läuft und man kann auf das aktuelle Spiel zugreifen.

14.5.1.2.3 Subpackage Player Im Package Player sind alle verschiedenen Spielertypen abgelegt. Alle implementieren die Methode "makeMove", welche aus dem Game aufgerufen wird. Die Methode "makeMove" gibt dem Game die Spalte zurück, oder eine Timeout- oder Abort-Exception falls ein Spieler zu lange für eine Zugeingabe braucht.

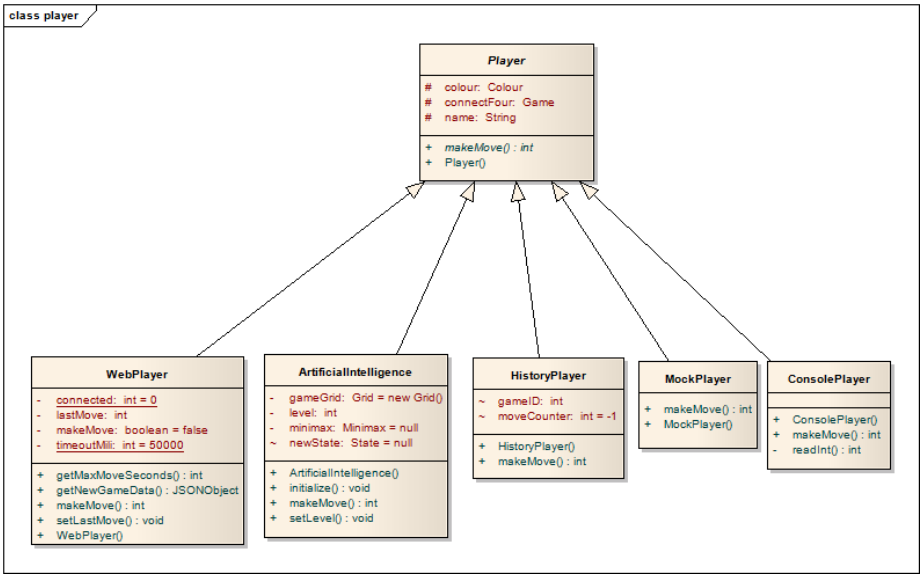


Abbildung 38: Package player

Der History Player holt vom gespeicherten Spiel bei "makeMove" immer den nächsten Zug aus der Datenbank. In der ArtificialIntelligence Klasse wird bei "makeMove" der Minimax Algorithmus gestartet, um den nächsten Zug zu evaluieren. Der MockPlayer und der ConsolePlayer dienen zu Testzwecken.

14.5.1.2.4 Subpackage Ki Dieses Package enthält die Dateien, welche für den Minimax Algorithmus benötigt werden.

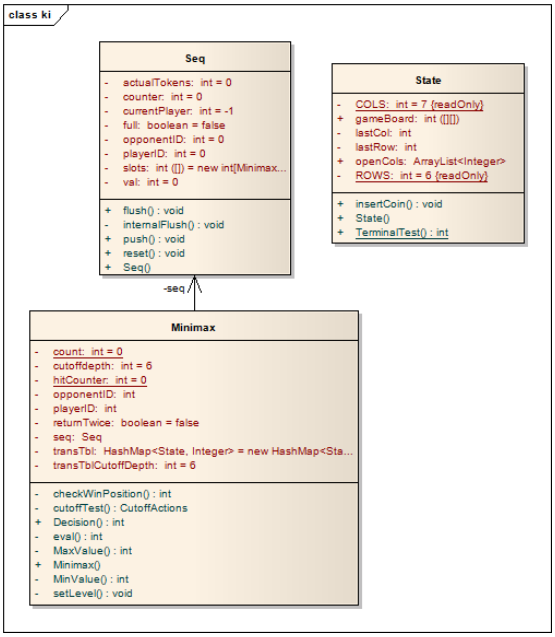


Abbildung 39: Package ki

14.5.1.2.5 Subpackage Robot In der Klasse Robot ist die Schnittstelle zum Roboter implementiert. Der Roboter wird über eine RS232 Schnittstelle angesteuert. Für die Kommunikation wird die Klasse SerialPort von RXTX[url02] gebraucht. Diese benutzt eine C Library um Daten über die COM Schnittstelle zu senden. Damit das Senden nach einer gewissen Zeit abgebrochen werden kann, wurde dies in die Klasse SerialWriter ausgelagert.

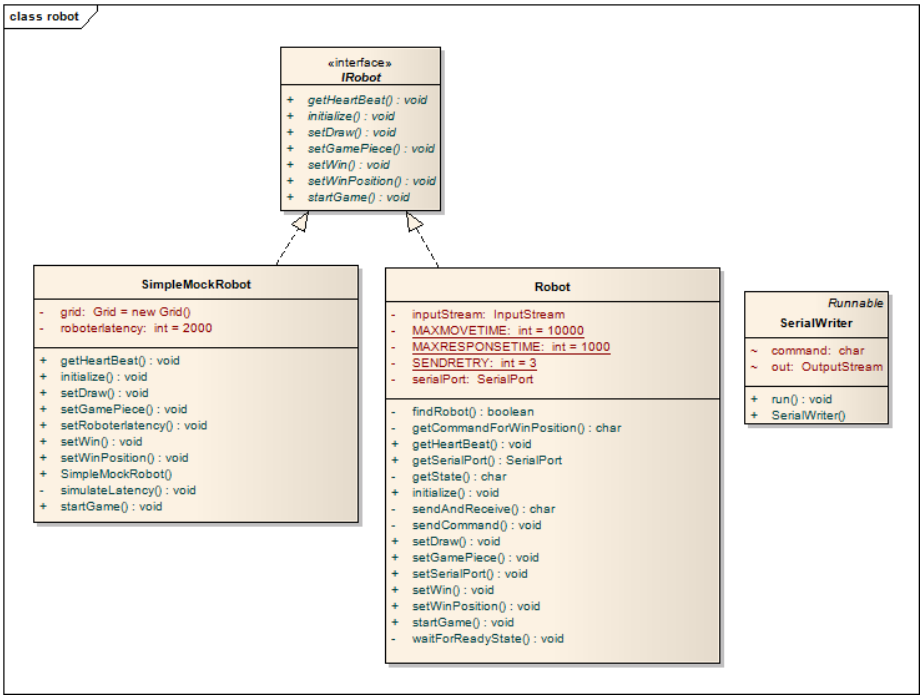


Abbildung 40: Package robot

Die Commands, die an den Roboter gesendet werden und die Antworten, die erhalten werden können, sind im Kapitel Schnittstellenbeschreibung (14.3) beschrieben. Damit das Progamm auch ohne den Roboter getestet werden kann und auch weil es den Roboter noch nicht gibt, wird das Interface IRobot auch noch von der Klasse SimpleMockRobot implementiert. Diese gibt die Befehle auf der Konsole aus. Um das GUI zu testen, kann im Mock Roboter eine Verzögerung eingebaut werden, da der wirkliche Roboter auch lange haben wird, um den Spielstein zu setzen. Mit der Methode "setRoboterlatency" kann festgelegt werden, wie lange die Verzögerung sein soll. Für UnitTests wird diese natürlich auf 0 gesetzt. Welche Roboterimplementation gewählt wird, kann im Config File gesetzt werden.

14.5.1.2.6 Subpackage GameEvents Jede Aktion, die während einem Spiel stattfinden kann, wird in einem Stack abgelegt, welcher Klassen die IGameEvent implementieren enthalten kann. Die GameEvents müssen alle in eine JSON Repräsentation umgewandelt werden, damit sie während dem Spiel dem User über AJAX gesendet werden können. Im GUI werden diese JSON Objekte dann von JavaScript Methoden verarbeitet.

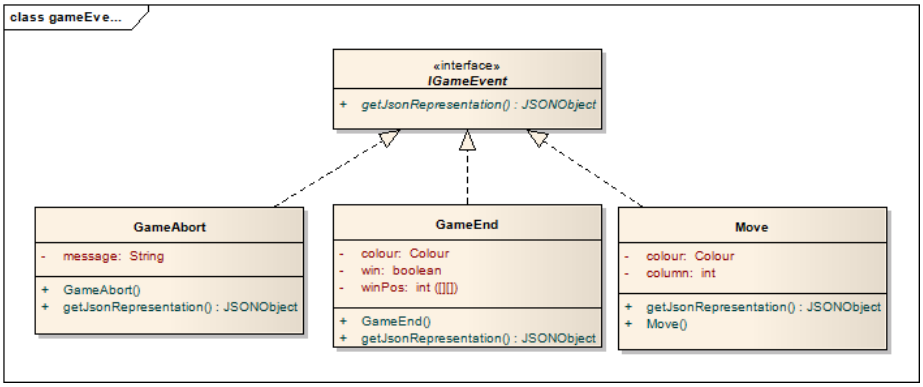


Abbildung 41: Package gameEvents

14.5.1.2.7 Subpackage Exception Um verschiedene Exceptions im Programm unterscheiden zu können, wurden verschiedene Exception Klassen erstellt, die alle von Exception erben und eine eigene Message enthalten. Exceptions können vom Grid (MoveNotAllowedException), vom Player (TimeoutException, UserAbortException) und vom Roboter (RobotException) geworfen werden. Die RobotException wird im GameContext gefangen, falls der Roboter gar nicht verfügbar ist. Alle Exceptions können während einem Game auftreten und werden im Main Loop des Games abgefangen. Das Game wird dadurch automatisch gestoppt und es wird dem User ein GameAbort Event gesendet.

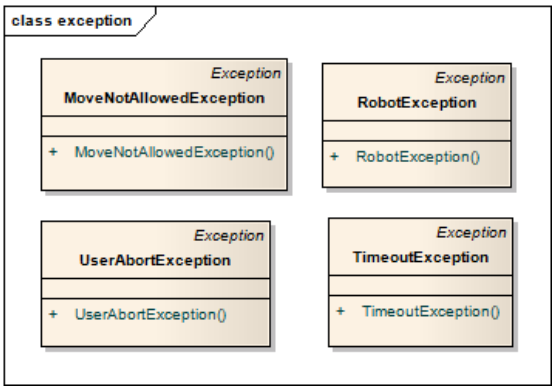


Abbildung 42: Package exception

14.5.1.2.8 Subpackage Config Das Package config enthält den ConnectFourLogger und den Zugriff auf das SettingsFile. Der ConnectFourLogger kappselt einen Log4J Logger und öffnet für diesen eine Konsole und einen FileAppender. Welches Level gerade geloggt wird, kann im Settingsfile eingestellt werden. Die Klasse Config regelt den Zugriff auf das Settingsfile. Im Settings-

file können Key / Value Paare gespeichert werden, die von der Applikation gebraucht werden.

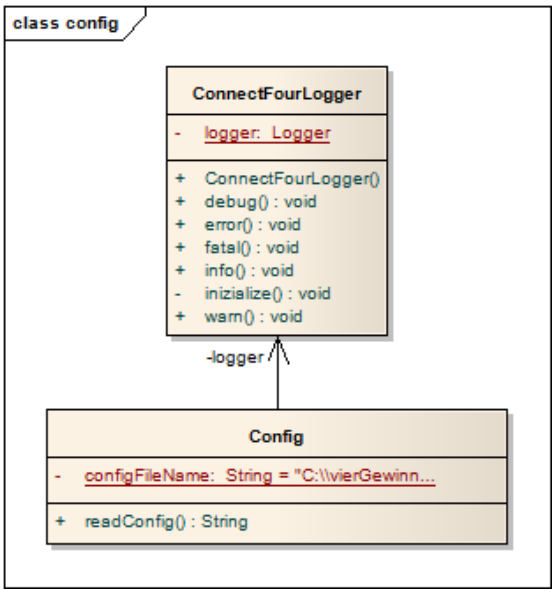


Abbildung 43: Package config

14.5.1.3 Package persistence

14.5.1.3.1 Beschreibung Die Datenhaltung ist sehr einfach gehalten. In der Datenhaltung werden lediglich alte Spiele gespeichert, damit diese wieder gezeigt werden können, wenn lange kein richtiges Spiel stattfindet. Sollte aus irgend einem Grund die Datanbank einmal nicht verfügbar sein, muss die Software normal weiter funktionieren.

14.5.1.3.2 Technologieentscheid Da nur sehr rudimentäre Datenbankfunktionen gebraucht werden, haben wir uns für eine leichtgewichtige SQLite Datenbank entschieden. Diese besteht lediglich aus einem einzelnen File, welches erstellt wird um die Daten zu speichern. Auf dieses File wird dann über JDBC direkt zugegriffen. Da die Ausführungsumgebung bei lokalen Tests und auf dem Tomcat auf verschiedenen Computern unterschiedlich ist, muss der Pfad zur Datenbank absolut angegeben werden. Damit dies einfach möglich ist, wird die Datenbank einfach im root Verzeichnis abgelegt.

14.5.1.3.3 Schema Das DB-Schema besteht aus zwei Tabellen. In der Tabelle Game wird nach Beendigung eines Spiels einen Eintrag gemacht, mit dem Spielausgang und den verschiedenen Spielern. In der Tabelle Move werden

die einzelnen Spielüge abgespeichert. Diese gehören immer zu einem Spiel und sind durchnummeriert. So kann ein Spiel später exakt nachgespielt werden.

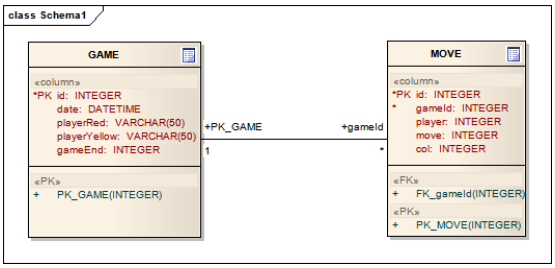


Abbildung 44: Datenbankschema

14.5.1.3.4 Klassendiagramm Die Persistenz-Schicht besteht nur aus der Klasse GameDb. Diese enthält Public Methoden um ein Game zu speichern, ein zufälliges Game aus der Datenbank zu lesen, sowie den nächsten Zug des aktuellen Historygames aus der DB zu holen. Die GameDb Klasse besteht nur aus Static Methoden, da kein Zustand benötigt wird.

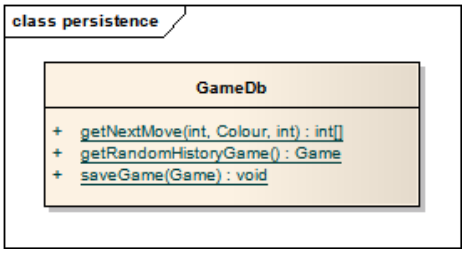


Abbildung 45: Klassendiagramm Persistenz

14.6 Liste App-Features und Supporting Arch-Features

In diesem Kapitel wird kurz und übersichtlich aufgelistet, welche Features der Applikation mit welcher Technologie implementiert wurden.

App-Features	Supportin Arch-Feature
Anzeige Spielfeld	HTML5, Canvas, Javascript
Übermittlung Spieldaten	AJAX Comet
Übermittlung aktuelle Warteliste	AJAX
Internet Seite generell	JSP, HTML5
Verbindung zum Roboter	RXTX Connection Library
History Play	SQLite Datenbank JDBC
Künstliche Intelligenz	Minimax Algorithmus mit Alpha-Beta-Pruning
Designunterscheidung iPad/iPhone	CSS Media-Queries
Desing UI	CSS3

15 Testdokumentation

15.1 Usability Tests / GUI Tests

15.1.1 Beschreibung

Bis jetzt wurde die Usability nur von uns getestet. Bei verschiedenen Demos wurden uns auch noch Inputs von anderen Usern gegeben, welche während der Demo die Navigation und Zugeingabe ausprobiert haben. Während den Meetings mit Prof. Dr. Markus Stolze wurden die Demo's auch von ihm getestet. Die Resultate dieser Tests wurden bis anhin in den Meetingsprotokollen festgehalten. Weitere Usability-Tests anhand TestCases werden noch durchgeführt.

15.1.2 Testabdeckung

Mit den Usability Tests wird die Benutzerfreundlichkeit der Bedienung des GUI getestet.

15.1.3 TestLog

Der nachfolgende Testcase wurde an folgenden Daten durchgeführt.

Datum:	Tester:	Testperson:	Bemerkungen:
19.12.2010	AG	Student der gerade seine SA macht.	Test hat funktioniert. Benutzer konnte alles wie geplant durchführen. Der Test fand allerdings noch nicht auf der Internetseite vierGewinnt.hsr.ch, sondern lokal auf einem Laptop statt.
21.12.2010	AG	Student der gerade seine SA macht.	Test hat funktioniert. Benutzer konnte alles wie geplant durchführen. Das zweite Klicken um den Zug zu bestätigen hat ihn irritiert. Als er sich in die Warteliste einreihen musste, hat er kurz nicht ganz verstanden warum er nun ein Spiel von anderen Spielern sieht. Der Test wurde als Video aufgezeichnet und kann im Ordner 08_Test auf der CD gefunden werden.

15.1.4 TestCase

Test:	Erwartetes resultat:
Nimm dein iPhone, Android Phone oder unser iPad hervor und starte es.	User hat ein Gerät, um auf die Seite zu verbinden.
Verbinde dich mit dem Wireless "HSR-VierGewinnt"	User geht in die Einstellungen seines Mobiltelefon und wählt das entsprechende Wireless-Netzwerk aus.
Öffne den Browser und verbinde dich mit viergewinnt.hsr.ch	User sieht Startseite des Programms.
Starte ein Einzelplayerspiel gegen den Roboter.	User startet ein Spiel. User gibt ohne Hinweis einen Benutzernamen ein und wählt den Schwierigkeitsgrad des Roboters.
Brich das Spiel nach fünf Zügen ab.	User klickt auf den "Abbrechen"-Knopf um das Spiel abbrechen.
Starte ein 2-Spieler Spiel und warte bis der Experte dem Spiel beitritt.	User versteht wie er ein 2-Spieler Spiel starten kann.
Spiele dieses Spiel ganz durch, bis du gewinnst, verlierst oder das Spiel unentschieden endet	User kann auf alle Situationen im Spiel richtig reagieren.
Warte bis der Experte ein Spiel startet und starte dann selber ein Spiel. Erkläre dabei, ob du alle Schritte verstehst.	User versteht warum er kein Spiel starten kann und trägt sich korrekt in die Warteliste ein.

15.2 System Tests

15.2.1 Beschreibung

In den Systemtests wird das korrekte Verhalten des GUIs in verschiedenen Situationen getestet. Diese Tests dienen insbesondere dazu, zu überprüfen ob alle Zustände abgedeckt sind und immer die richtigen Webseiten dem User angezeigt werden.

15.2.2 Testabdeckung

Diese Tests testen die Funktionalität des GUIs, da diese nicht sinnvoll mit Unit Tests überprüft werden kann.

15.2.3 TestDurchführung

Der Test wurde am 19.12. von AG durchgeführt.

Test:	Erwartetes Resultat:	Resultat:
Ein einzelner Benutzer kann ein Singleplayerspiel eröffnen und seinen Namen und die Spielstärke der KI einstellen.	Funktionierendes Spielfeld mit Anzeige des richtigen Namen und des KI Levels.	wie beschrieben
Spieler 1 startet ein Singleplayerspiel. Spieler 2 besucht die Webseite und will spielen.	Er kann sich korrekt in der Warteliste eintragen und sieht das aktuelle Spiel. Sobald das laufende Spiel fertig ist wird Spieler 2 aufgerufen. Nach einer Bestätigung wird automatisch ein Singleplayerspiel gegen die KI eröffnet.	wie beschrieben
Wie vorheriger Test aber mit zwei weiteren Spieler 3 und 4. Nach Beendigung von Spiel 1 starten Spieler 2 gegen Spieler 3 ein weiteres Spiel und Spieler 4 bleibt in der Warteliste.	Sobald das Spiel fertig ist wird zuerst der Spieler 2 und kurz darauf der Spieler 3 gefragt ob sie spielen wollen. Es wird ein Multiplayer Spiel eröffnet und bei Spieler 4 wird dieses Spiel angezeigt. Die Warteliste zeigt die erwarteten Positionen.	wie beschrieben
Laufendes Spiel abbrechen.	Durch klicken auf den Abbrechen Button wird das Spiel mit einer UserAbortMessage abgebrochen.	wie beschrieben

Laufendes Spiel durch Timeout abbrechen.	Nach 50s wird das laufende Spiel mit einer TimeoutMessage abgebrochen und es kann ein neues Spiel gestartet werden. Die letzten 15s wird dem User ein cCountdown angezeigt.	wie beschrieben

15.3 Unit Tests

15.3.1 Beschreibung

Für alle wichtigen Methoden im Domain Layer wurden Unit Tests mit JUnit 4 erstellt. Auf die Überprüfung ob Getter und Setter die richtigen Resultate liefern wird verzichtet, da diese jeweils generiert wurden und so trivial sind, dass in ihnen keine Fehler vorkommen können.
Um alle Tests gemeinsam laufen zu lassen gibt es die Klasse AllTests im Package "test", an welche alle Unit Tests angehängt sind.

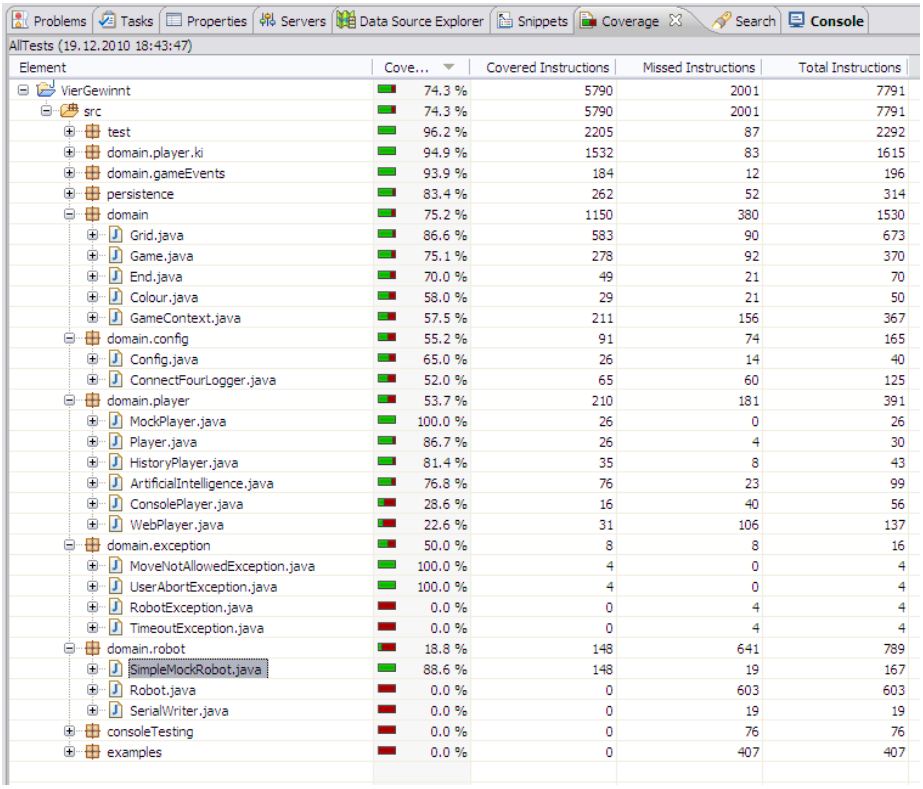
15.3.2 Mock Objekte

Damit eine kurze Laufzeit und unabhängige Tests möglich sind wurden zwei Mock Objekte erstellt.

- Die Klasse MockPlayer ist ein extrem simpler Spieler, welcher seinen Spielstein zufällig in eine Spalte wirft. So kann die Game-Klasse ohne lange Laufzeiten getestet werden.
- Die zweite Mock Klasse ist der SimpleMockRobot. Dieser repräsentiert den Roboter und gibt die Befehle an den Roboter in ein Logfile aus. Der richtige Roboter ist momentan noch nicht vorhanden und wird später ziemlich lange haben um einen Zug durchzuführen und kann deshalb nicht bei den Tests eingesetzt werden.

15.3.3 Testabdeckung

Die Testabdeckung wurde durch das Tool EclEmma überprüft. Printscreen der letzten Überprüfung:



Element	Cove...	Covered Instructions	Missed Instructions	Total Instructions
VierGewinnt	74.3 %	5790	2001	7791
src	74.3 %	5790	2001	7791
test	96.2 %	2205	87	2292
domain.player.ki	94.9 %	1532	83	1615
domain.gameEvents	93.9 %	184	12	196
persistence	83.4 %	262	52	314
domain	75.2 %	1150	380	1530
Grid.java	86.6 %	583	90	673
Game.java	75.1 %	278	92	370
End.java	70.0 %	49	21	70
Colour.java	58.0 %	29	21	50
GameContext.java	57.5 %	211	156	367
domain.config	55.2 %	91	74	165
Config.java	65.0 %	26	14	40
ConnectFourLogger.java	52.0 %	65	60	125
domain.player	53.7 %	210	181	391
MockPlayer.java	100.0 %	26	0	26
Player.java	86.7 %	26	4	30
HistoryPlayer.java	81.4 %	35	8	43
ArtificialIntelligence.java	76.8 %	76	23	99
ConsolePlayer.java	28.6 %	16	40	56
WebPlayer.java	22.6 %	31	106	137
domain.exception	50.0 %	8	8	16
MoveNotAllowedException.java	100.0 %	4	0	4
UserAbortException.java	100.0 %	4	0	4
RobotException.java	0.0 %	0	4	4
TimeoutException.java	0.0 %	0	4	4
domain.robot	18.8 %	148	641	789
SimpleMockRobot.java	88.6 %	148	19	167
Robot.java	0.0 %	0	603	603
SerialWriter.java	0.0 %	0	19	19
consoleTesting	0.0 %	0	76	76
examples	0.0 %	0	407	407

Abbildung 46: EclEmma Test Coverage

Die Zahl der Abdeckung von 74% erscheint im ersten Augenblick etwas niedrig. Bei genauerem hinschauen sieht man jedoch, dass dies so ist, weil auch Beispiele und Code für das manuelle Testen in dieser Zahl enthalten ist. Zudem ist auch der Code der Robot Klasse enthalten, welche extra für die Tests gemockt wurde und wie im Kapitel Schnittstellentest (15.5) beschrieben separat getestet wird. Gleiches gilt auch für die Klasse WebPlayer, die ebenfalls gemockt wurde.

15.3.4 TestLog

Unit Tests wurden während des ganzen Projekts fortlaufend erweitert und bei jeder Änderung ausgeführt, um zu überprüfen, ob alle Funktionen noch richtig funktionieren. Auf der Abbildung sieht man eine Durchführung der Tests:

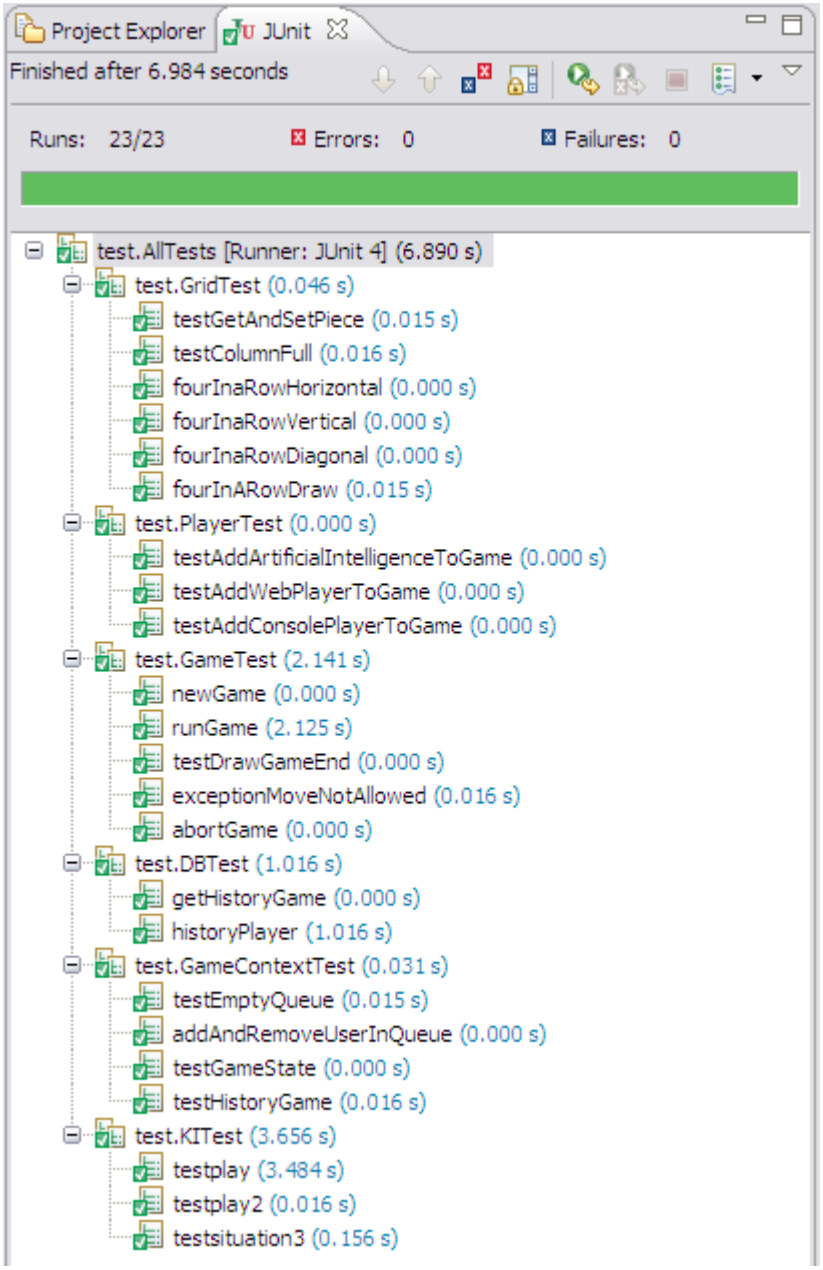


Figure 47: Unit Test durchführung Stand 19.12.2010

15.4 Manuelle Tests

15.4.1 Beschreibung

Während des Projektes wurden die verschiedenen Funktionen regelmässig lokal manuell getestet. Die verschiedenen Main Methoden wurden alle in separate Klassen ins Package consoleTesting verschoben.

In der Klasse ConsoleGame gibt es eine Main Methode mit der man ein Konsolenspiel starten kann. Dafür gibt es die Klasse ConsolePlayer, welche die Zugeingabe über die Konsole implementiert. Durch diese Methode und Klasse ist das lokale Debugging viel einfacher möglich, als wenn man dafür extra den Tomcat starten muss und über den Browser spielt.

In der Klasse ConsoleRobot gibt es die Main Methode des Roboters über die man das Senden einzelner Befehle an den Roboter direkt ausführen kann.

15.4.2 Testabdeckung

Mit den manuellen Tests wurde die Grundfunktionalität des Spiels über die Konsole getestet. Des weiteren wurden verschiedene Elemente des GUI manuell getestet.

15.4.3 TestCase Manual Console Play

Die folgende Abbildung zeigt einen Ausschnitt von einem Vier Gewinnt Spiel auf der Konsole. Dieser Modus wird nur zu Testzwechen gestartet.

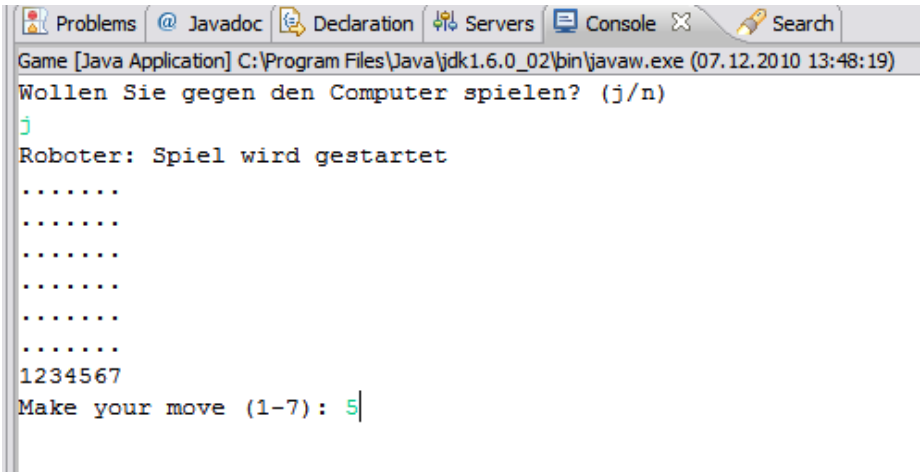


Abbildung 48: Ausschnitt Vier Gewinnt auf der Konsole spielen

15.5 Schnittstellen Tests

15.5.1 Beschreibung

Der Roboter ist mit dem Server über eine serielle Schnittstelle verbunden. Da die Teile der Maschinentechnik sowie der Elektrotechnik in diesem Semester noch nicht fertig sind, ist ein kompletter Test der Schnittstelle leider nicht möglich.

Um wenigstens ein Teil davon zu testen hat Renato Müller die SPS (Speicherprogrammierbare Steuerung), das ist der Teil mit welchem der Server kommuniziert soweit fertig gestellt, das schon mal ein einzelner Befehl "AskStatus" vom Server an den Roboter gesendet werden konnte und der Roboter diesen mit einem "Ready" beantworten konnte.

15.5.2 Testaufbau

Der Test war folgendermassen aufgebaut: Auf einem Laptop lief unser Serverprogramm. Am Laptop war an der USB Schnittstelle das USB-to-Serial Kabel eingesteckt. Auf der anderen Seite wurde das Kabel am Serial Modul der SPS eingesteckt. Die SPS war zudem an einem PC angeschlossen, wo man die Zustände der einzelnen Bausteine anschauen konnte.

Abbildung 49: SPS mit eingestecktem Serial Kabel

Abbildung 50: USB-Serial Verbindungskabel

15.5.3 Testabdeckung

Da auf der Seite der SPS das Programm noch nicht vollständig vorhanden ist, konnte nicht die gesamte Kommunikation zwischen dem Server und dem Roboter getestet werden.

15.5.4 TestLog vom 17.12.2010

Test:	Erwartetes Resultat:	Resultat:
Server starten wenn im Settingsfile ROBOT_MODE=REAL gesetzt wurde.	Server startet ohne Fehlermeldung	wie erwartet
Auf Webseite zugreifen, damit das Application Bean GameContext erstellt wird und der Server Main Loop der Application gestarted wird.	Webseite zeigt Roboter-Error an, da dieser noch nicht initialisiert ist. Initialisierung wird gestartet	wie erwartet
Initialisierung des Roboters:		

1. Allen verfügbaren COM Ports wird eine “AskStatus” (A) Meldung gesendet, um herauszufinden an welchem der Roboter anhängt ist.	Es werden 3 COM-Ports überprüft und bei den ersten zwei kommt eine Meldung, dass der Roboter nicht angeschlossen ist. Beim dritten COM-Port wird der Roboter erkannt.	wie erwartet. Siehe Abbildung 51
2. Roboter wird mit dem Befehl “B” initialisiert.	Dies wurde abgestellt, da die SPS dies noch nicht implementiert	–
3. Server sendet im HeardBeat Loop alle 5 Sekunden dem Roboter eine “AskStatus” (A) Message und erhält eine “Ready” (1) Antwort vom Server	Konsolenausgabe der gesendeten Meldungen	wie erwartet. siehe Abbildung 52
4. Auf der Webseite ein Spiel starten.	Konsolenausgabe der gesendeten Meldung “C”. Da der Roboter dies noch nicht implementiert wird senden nach 3 versuchen abgebrochen und in der Applikation wird eine RobotException geworfen. Das Spiel wird aufgeräumt.	wie erwartet. siehe Abbildung 53
5. Server sendet im HeardBeat Loop wieder alle 5 Sekunden dem Roboter eine “AskStatus” (A) Message und erhält eine “Ready” (1) Antwort vom Server	Da der Server nun wieder ein “Ready” vom Roboter erhällt sollte Spielen wieder normal möglich sein.	wie erwartet. siehe Abbildung 54

```
INFO  17 12 2010 10:23:30 : Initialize Robot out of main loop
DEBUG 17 12 2010 10:23:30 : initialize the Robot out of the gameContext
INFO  17 12 2010 10:23:30 : Take the real Robot
INFO  17 12 2010 10:23:31 : Try to connect to: COM3
DEBUG 17 12 2010 10:23:31 : Try sending: A to the Robot
INFO  17 12 2010 10:23:31 : Received Return Code: A from the Robot (Byte:65)
INFO  17 12 2010 10:23:31 : The robot is not connected to that port
INFO  17 12 2010 10:23:31 : Try to connect to: COM5
DEBUG 17 12 2010 10:23:31 : Try sending: A to the Robot
INFO  17 12 2010 10:23:32 : Maximal retries not reached try to resend request
DEBUG 17 12 2010 10:23:32 : Try sending: A to the Robot
INFO  17 12 2010 10:23:33 : Maximal retries not reached try to resend request
DEBUG 17 12 2010 10:23:33 : Try sending: A to the Robot
WARN  17 12 2010 10:23:35 : Maximal retries reached. Return X as Error State
INFO  17 12 2010 10:23:35 : The robot is not connected to that port
INFO  17 12 2010 10:23:35 : Try to connect to: COM7
DEBUG 17 12 2010 10:23:35 : Try sending: A to the Robot
INFO  17 12 2010 10:23:35 : Received Return Code: 1 from the Robot (Byte:49)
DEBUG 17 12 2010 10:23:35 : Found robot
```

Abbildung 51: Logeinträge von Roboterinitialisierung

```
DEBUG 17 12 2010 10:23:35 : Try sending: A to the Robot
INFO  17 12 2010 10:23:35 : Received Return Code: 1 from the Robot (Byte:49)
INFO  17 12 2010 10:23:35 : Received command 1 from robot.
INFO  17 12 2010 10:23:35 : Check Heard Beat of Robot out of main loop
DEBUG 17 12 2010 10:23:35 : Try sending: A to the Robot
INFO  17 12 2010 10:23:35 : Received Return Code: 1 from the Robot (Byte:49)
INFO  17 12 2010 10:23:45 : Check Heard Beat of Robot out of main loop
DEBUG 17 12 2010 10:23:45 : Try sending: A to the Robot
INFO  17 12 2010 10:23:45 : Received Return Code: 1 from the Robot (Byte:49)
INFO  17 12 2010 10:23:55 : Check Heard Beat of Robot out of main loop
DEBUG 17 12 2010 10:23:55 : Try sending: A to the Robot
INFO  17 12 2010 10:23:56 : Received Return Code: 1 from the Robot (Byte:49)
```

Abbildung 52: Logeinträge HeardBeat

```
DEBUG 17 12 2010 10:23:58 : Try sending: C to the Robot
INFO  17 12 2010 10:23:59 : Maximal retries not reached try to resend request
DEBUG 17 12 2010 10:23:59 : Try sending: C to the Robot
INFO  17 12 2010 10:24:00 : Maximal retries not reached try to resend request
DEBUG 17 12 2010 10:24:00 : Try sending: C to the Robot
WARN  17 12 2010 10:24:01 : Maximal retries reached. Return X as Error State
ERROR 17 12 2010 10:24:01 : Leider hat der Roboter eine StörungError:
                            The sending of the command C to the robot did not Work correctly
```

Abbildung 53: Logeinträge von Spiel starten

```
INFO  17 12 2010 10:24:06 : Check Heard Beat of Robot out of main loop
DEBUG 17 12 2010 10:24:06 : Try sending: A to the Robot
INFO  17 12 2010 10:24:06 : Received Return Code: 1 from the Robot (Byte:49)
INFO  17 12 2010 10:24:16 : Check Heard Beat of Robot out of main loop
DEBUG 17 12 2010 10:24:16 : Try sending: A to the Robot
INFO  17 12 2010 10:24:16 : Received Return Code: 1 from the Robot (Byte:49)
INFO  17 12 2010 10:24:26 : Check Heard Beat of Robot out of main loop
DEBUG 17 12 2010 10:24:26 : Try sending: A to the Robot
INFO  17 12 2010 10:24:26 : Received Return Code: 1 from the Robot (Byte:49)
```

Abbildung 54: Logeinträge von HeardBeat2

16 Benutzeranleitung

16.1 Anleitung für Infotag Besucher

Damit du Vier Gewinnt spielen kannst, musst du dich mit dem Wi-Fi Netzwerk "HSR-VierGewinnt" Verbinden.

Falls du ein iPhone hast machst du dies wiefolgt: Gehe ins Menü "Einstellungen -> Wi-Fi" dort musst du nur das Wi-Fi Aktivieren und dann das Netzwerk "HSR-VierGewinnt" auswählen. Bei einem Android Phone funktioniert dies praktisch gleich.

Figure 55: iPhone Wi-Fi einstellungen

Nun musst du nur noch den Browser (Safari) öffnen und die Internetseite "viergewinnt.hsr.ch" öffnen.

Auf der Internetseite ist alles weitere beschrieben.

16.1.1 Spielregeln

Das klassische Brettspiel wird auf einem senkrecht stehenden hohlen Spielbrett gespielt, in das die Spieler abwechselnd ihre Spielsteine fallen lassen. Das Spielbrett besteht aus sieben Spalten (senkrecht) und sechs Reihen (waagerecht). Jeder Spieler besitzt 21 gleichfarbige Spielsteine. Wenn ein Spieler einen Spielstein in eine Spalte fallen lässt, besetzt dieser den untersten freien Platz der

Spalte. Gewinner ist der Spieler, der es als erster schafft, vier oder mehr seiner Spielsteine waagerecht, senkrecht oder diagonal in eine Linie zu bringen. Das Spiel endet unentschieden, wenn das Spielbrett komplett gefüllt ist, ohne dass ein Spieler eine Viererlinie gebildet hat.[url03]

16.2 Anleitung für den Informationsdienst

Diese Anleitung erklärt kurz und prägnant was gemacht werden muss, um den Server an einem Infotag in Betrieb zu nehmen.

1. Server und wireless Accesspoint miteinander verkabeln.
2. Den Roboter (die SPS) mit dem seriellen Kabel und dem Serial-to-USB Adapter mit dem Server verbinden.
3. Den Server und den Accesspoint starten.
4. Auf dem Server muss man sich einloggen, desshalb braucht man auch einen Bildschirm, Tastatur und eine Maus.
   (a) Benutzername: Administrator
   (b) Passwort: HSR-VierGewinnt
5. Sobald der Server fertig hochgefahren ist, kann der Roboter gestartet werden (separate Anleitung).
6. Jetzt kann mit dem iPad oder auf dem Server im Browser die Webseite viergewinnt.hsr.ch geöffnet werden. (Siehe separate Anleitung Kap. 16.1)
   (a) Achtung: Beim ersten Öffnen der Seite nach einem Neustart wird der Roboter initialisiert. Dies kann einige Momente dauern. Auf der Internetseite wird dabei eine Errormeldung angezeigt, welche automatisch verschwindet, sobald der Roboter bereit ist.

16.2.1 Fehlerbehebung:

- Sollte der Roboter oder sonst etwas nicht funktionieren kann vielleicht ein Blick ins Logfile helfen. Dieses findet man unter "C:\vierGewinnt\vierGewinnt.log"
- Wenn auf der Internetseite alles funktioniert, aber der Robter nichts macht, kann es sein, dass im Settingsfile die Ausgabe auf den virtuellen Testroboter eingestellt ist.
  – Dies findet man unter "C:\vierGewinnt\vierGewinnt.properties"
- Eventuell kann auch mal ein Neuladen der gesamten Applikation etwas bringen. Dazu muss man den Tomcat Manager starten (Lokale Internetseite, link ist auf Desktop)

  - Benutzername ist admin, Passwort: viergewinnt
  - Dort kann man das Projekt VierGewinnt reloaden (Neustarten)
  - Das War File des Projekts ist unter "C:\vierGewinnt\" abgelegt.

16.3 Anleitung für einen Administrator (Neu aufsetzen)

Falls ein neuer Server für das VierGewinnt Projekt aufgsetzt wird, müssen folgende Punkte speziell beachtet werden.

- Das Projekt wurde unter Windows Server 2008 R2 getestet.
- Router muss geflashed werden. Anleitung unter [url04].
  - DNS eintragen: Statische IP des Servers. (Router muss wissen, dass der Server ein DNS ist -> Alle Anfragen an den Server weiterleiten)
  - Portforwarding von 80 auf 8080
- Auf dem Server muss ein DNS Programm laufen, damit die Webseite viergewinnt.hsr.ch lokal aufgelösst werden kann.
  - DNS Manager ist bereits auf dem Windows Server 2008 drauf und muss nur noch aktiviert werden. Installationsanleitung unter [url05].
  - viergewinnt.hsr.ch muss als Forward Lookup Zone eingerichtet werden und auf IP des Servers zeigen.
- Auf dem Server muss Java und Tomcat installiert werden
- Für das USB-to-Serial Kabel braucht es einen Treiber. Treiber für verschiedene Windows Betriebssysteme sind auf der abgegebenen CD im Ordner Schnittstelle.
- Für die Ansteuerung der seriellen Schnittstelle aus Java ist die RXTX Library nötig. Installationsanleitung und Files für Windows sind auf der abgegebenen CD im Ordner 15_Schnittstellen_Treiber.

16.4 Anleitung für Programmierer (Entwicklungsumgebung)

Damit am Projekt weiterentwickelt werden kann, müssen auch ein Teil der unter Administrator beschriebenen Tasks ausgeführt werden. Folgendes ist nötig, um den Code zu bearbeiten:

- Java EE installieren
- File -> Import -> WAR File. Das WAR File unseres Projekts welches auch den Source Code enhält, ist im Ordner 14_SourceCode auf der abgegebenen CD.

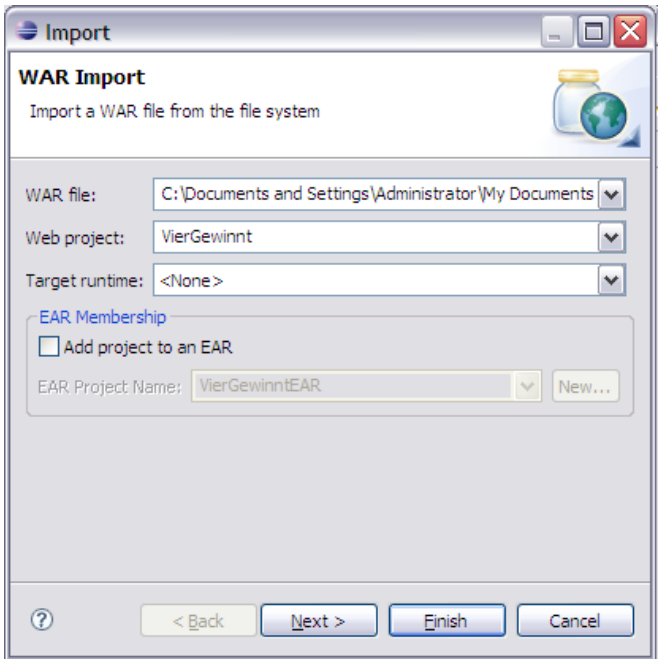


Figure 56: Auswahl WAR File

- Im BuildPath muss JUnit 4 noch hinzugefügt werden

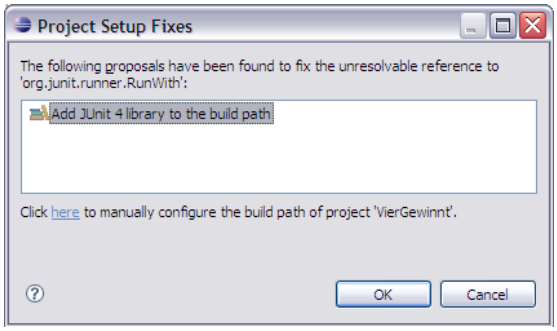


Figure 57: Fix Project Setup

- Im C:\ muss ein Ordner "vierGewinnt" erstellt werden. In diesen Ordner muss das Properties File kopiert werden, welches im Ordner SourceCode auf der abgegebenen CD zu finden ist.
  - Falls nicht auf einem Windows Rechner gearbeitet wird, kann im Code der Pfad angepasst werden. (Der Pfad muss in der Klasse GameDb, Config und ConnectFourLogger angepasst werden)

- Wie im Kapitel 16.3, Anleitung Administrator beschrieben muss nun noch die RXTX Library installiert werden.
- Nun sollte das Projekt keine Errors mehr aufweisen und alle Unit Tests sollten fehlerfrei laufen.
- Jetzt muss noch ein Tomcat dem Workspace hinzugefügt werden, damit das gesamte Projekt getestet werden kann.

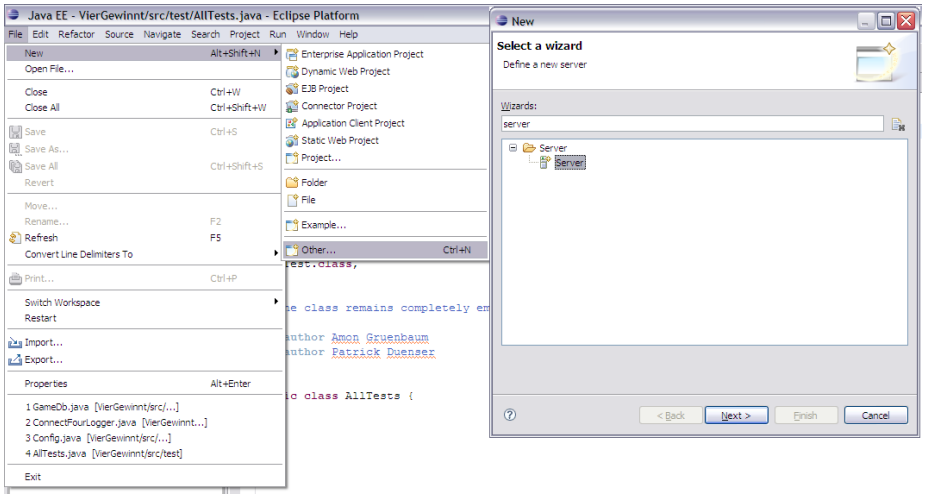


Figure 58: Tomcat Server in Eclipse erstellen

- Tomcat Version und Installationsverzeichnis auswählen

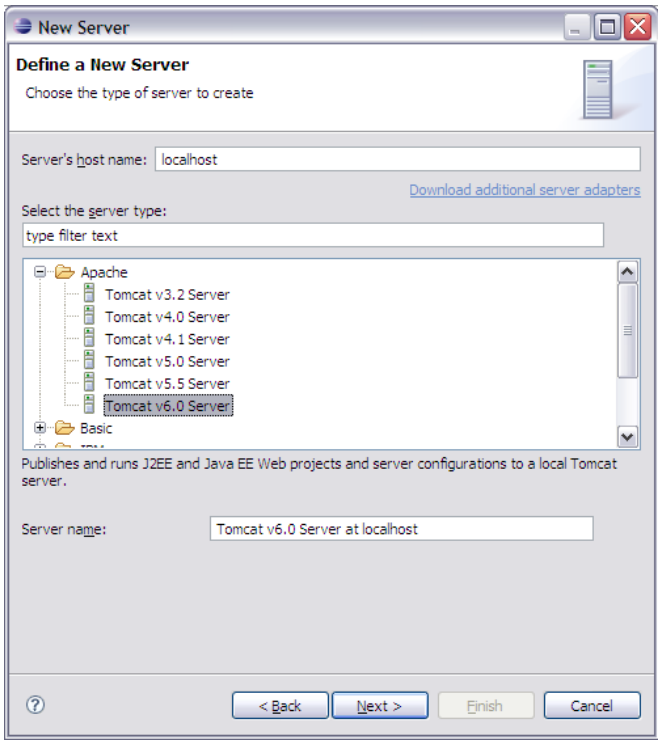


Figure 59: Auswahl des Servers

- Projekt auswählen und "Run on Server" anklicken.
- Mit Firefox, Safari oder Opera auf http://localhost:8080/VierGewinnt/ zugreifen.

Part IV
Anhang

17 Glossar

Begriff	Beschreibung
A* Algorithmus	Suchalgorithmus, der den kürzesten Pfad zwischen zwei Knoten berechnet.
Ajax	Asynchronous JavaScript and XML. Es bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Browser und dem Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, und die Seite zu verändern, ohne sie komplett neu zu laden.
Alpha-Beta-Pruning	Optimierte Version des Minimax Algorithmus bei dem gewisse Teile des Suchbaums nicht angeschaut werden, da sie nicht zu einem guten Ergebnis führen können.
Apache	Webserver. Der Apache HTTP Server ist eine freie Software.
Comet	Web Programmier Model (Pattern) bei dem eine HTTP-Verbindung zum Server geöffnet wird und so lange offen bleibt bis der Server Daten zum Client senden will. Dies wird auch Server Push genannt.
Cascading Style Sheets	Cascading Style Sheets ist eine deklarative Stylesheet-Sprache für strukturierte Dokumente, wie z. B. HTML.
Domain Name Service	Dienst für die Beantwortung von Anfragen zur Namensauflösung.
Hypertext Markup Language	Die Hypertext Markup Language ist eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten.
HTML5	HTML5 ist eine neue Spezifikation der HTML Websprache, die von der WHATWG (Web Hypertext Aplication Technology Working Group) ausgearbeitet wurde und als Gegenvorschlag zum trägen W3C (World Wide Web Consortium) gesehen werden kann.
Javadoc	Software-Dokumentationswerkzeug, welches aus Java Source Code HTML Dateien generiert, die alle Kommentare des Source Codes visuell darstellen.
LaTeX	LaTeX ist ein Softwarepaket, das die Benutzung des Textsatzprogramms TeX mit Hilfe von Makros vereinfacht.

Minimax	Minimax ist ein Algorithmus zur Ermittlung der optimalen Spielstrategie.
Nullsummenspiel	Nullsummenspiele beschreiben in der Spieltheorie Situationen, also Spiele im verallgemeinerten Sinne, bei denen die Summe der Gewinne/Verluste aller Spieler zusammengenommen gleich null ist.
OLE for Process Control	OLE for Process Control, standardisierte Software-Schnittstellen im Bereich Automatisierungstechnik.
Scrum	Vorgehensmodell der agilen Softwareentwicklung.
Speicherprogram-mierbare Steuerung	Gerät zur Steuerung einer Maschine oder Anlage, welches auf digitaler Basis programmiert wird.
Subversion	Versionsverwaltung von Dateien und Verzeichnissen.
(Graphical) User Interface	Grafische Benutzerschnittstelle einer Software-Anwendung .
Tomcat	siehe Apache
Universal Serial Bus	Universal Serial Bus ist ein serielles Bussystem zur Verbindung eines Computers mit externen Geräten.
Vier Gewinnt	Zweipersonen - Strategiespiel mit dem ziel als Erster vier der eigenen Spielsteine in eine Linie zu bringen.

18 Abkürzungsverzeichnis

CSS	Cascading Style Sheets
DNS	Domain Name Service
HTML	Die Hypertext Markup Language
OPC	OLE for Process Control
SPS	Speicherprogrammierbare Steuerung
SVN	Subversion
(G)UI	(Graphical) User Interface
USB	Universal Serial Bus

19 Abbildungsverzeichnis

List of Figures

20 Inhalt der CD

In den folgenden Abbildungen ist die Ordnerstruktur der CD ersichtlicht:

Figure 60: Ordnerstruktur der CD

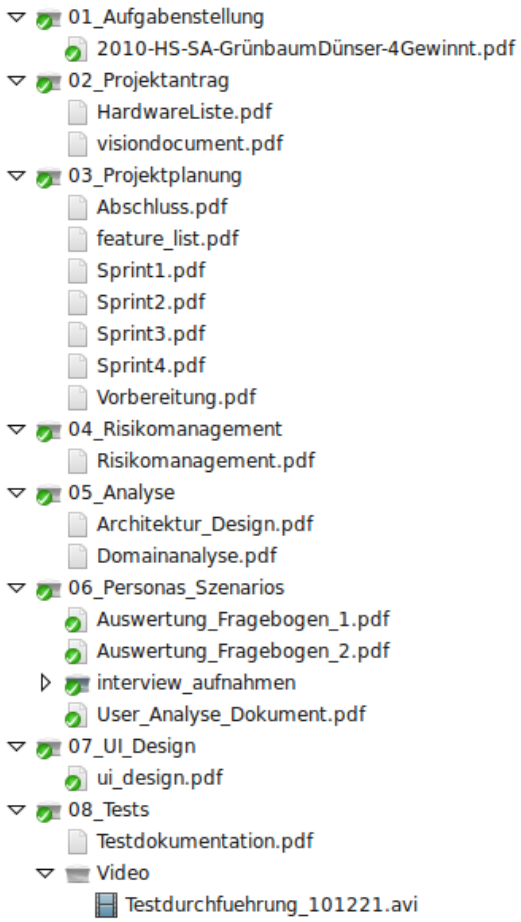


Figure 61: Dokumente auf der CD (Teil 1)

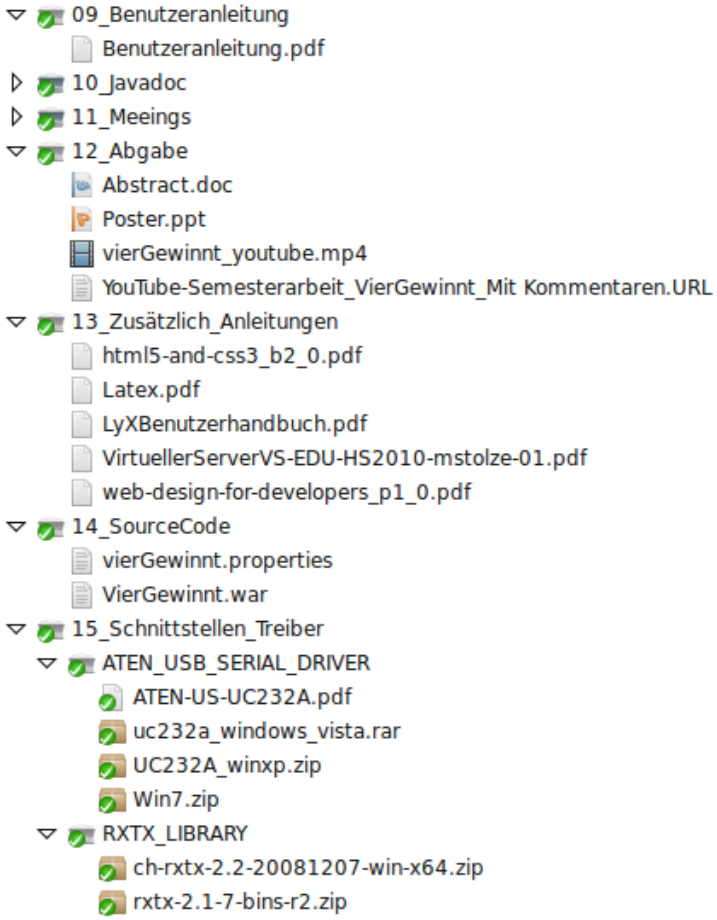


Figure 62: Dokumente auf der CD (Teil 2)

References

[Rustem02] Berç Rustem / Melendres Howe: Algorithms for Worst-case Design and Applications to Risk Management, Princeton 2002, 1-15

[Baier06] Hendrik Baier: Der Alpha-Beta-Algorithmus und Erweiterungen bei Vier Gewinnt, Offenbach 2006

[Kroener10] Peter Kröner: HMTL5 - Webseiten innovativ und zukunftsicher. München 2010, S. 249 - 304

[url01] http://de.wikipedia.org/wiki/Minimax-Algorithmus#Pseudocode (13.10.2010)

[url02] http://www.rxtx.org (10.11.2010)

[url03] http://de.wikipedia.org/wiki/Vier_gewinnt#Regeln (18.12.2010)

[url04] http://dd-wrt.com/wiki/index.php/Installation (21.12.2010)

[url05] http://blogs.techrepublic.com.com/datacenter/?p=327 (21.12.2010)

[url06] http://www.oracle.com/technetwork/java/codeconvtoc-136057.html (20.10.2010)

21 Zeiterfassung

Sprint Vorbereitung

20.09.10- 24.09.10

Ziel

Vorbereitung für SA, aufsetzen des Projekts

Nr.	Priorität	Status		Aufgabe	Verantwortlich	Soll Arbeitsstunden	Ist Arbeitsstunden		Ist ArbeitstundenTotal
							Woche 1		
1	Hoch	erledigt	**Administration**	SVN Aufsetzen	Amon	1	1		1
2	Hoch	erledigt	**Administration**	Wikipage vorbereiten	Amon	1	1		1
3	Hoch	erledigt	**Administration**	Meetings	Team	4	2	1	3
4	Hoch	laufend	**Administration**	SCRUM Projektplan	Team	8	2	2	4
5	Mittel	laufend	**Administration**	Projektantrag erstellen	Team	6	2	2	4
6	Mittel	erledigt	**Administration**	Ideensammlung	Team	2	2	0.5	2.5
7	Mittel	erledigt	**Administration**	Organisation	Team	2	3		3
8	Niedrig	erledigt	**Administration**	Prestudies	Team	4	3	2	5
9	Niedrig	erledigt	**User Analyse**	Szenarios erstellen	Team	1	0.5	0.5	1
10	Niedrig	erledigt	**User Analyse**	Personas erstellen	Team	2	1	1	2
11		erledigt	**Meilenstein**	SE Model festgelegt	Team				0
12		erledigt	**Meilenstein**	Projektplan	Team				0

Fertig: 10/12

Sprint Beendet

Soll Arbeitstunden Total:	31	Ist Arbeitstunden Total:	26.5
Verbleibende Arbeitsstunden:	4.5	Ist Arbeitsstunden Amon:	17.5
Arbeitsstunden pro Tag:	8.5	Ist Arbeitsstunden Patrick:	9

Sprint 1

27.09.2010 – 15.10.10

Ziel

4 Gewinnt als Mockup Object (abgenommen durch benutzer)

Arbeitsstunden pro Woche

Nr.	Priorität	Status		Aufgabe	Verantwortlich	Soll Arbeitsstunden	Ist Arbeitsstunden						Ist ArbeitstundenTotal
							Woche 1		Woche 2		Woche 3		
1	Hoch	erledigt	**Scrum Tasks**	Sprint planen	Team	1	0.5	0.5					1
2	Hoch	erledigt	**Scrum Tasks**	Spirnt review	Team	2					1	1	2
3	Hoch	erledigt	**Administration**	Projektantrag	Team	4	6	1	4	1	1		13
4	Hoch	erledigt	**Administration**	Risikomanagement	Team	8			5	2			7
5	Mittel	erledigt	**Administration**	Inhaltsverzeichnis	Team	1		1					1
6	Mittel	erledigt	**Administration**	Meilensteinplanung	Team	1	1						1
7	Mittel	laufend	**Domain Analyse**	Domainanalyse	Team	34	1		2	3	10	5	21
8	Mittel	Offen	**Architektur Design**	Algorithmus evalutaion	Team	7		3		3		1	7
9	Mittel	erledigt	**UI Design**	Paperprototyp Version 1	Team	4	2		2				4
10	Mittel	erledigt	**Domain Analyse**	Use Cases	Team	7				3	1	4	8
11	Mittel	laufend	**User Analyse**	Interview	Team	4					2	3	5
12	Mittel	laufend	**User Analyse**	Personas	Team	3		1		1			2
13	Mittel	laufend	**User Analyse**	Szenarios	Team	3		1	1	2			4
14	Mittel	erledigt	**Administration**	Selbststudium	Team	3		1	1	1			3
15	Mittel	erledigt	**Administration**	Meetings	Team	9	2	2	2	2	2	2	12
16	Hoch	erledigt	**Administration**	Dokumentvorlagen	Team	3	1	1					2
17	Hoch	erledigt	**Administration**	SprintBacklogVorlage	Team	4	1	1					2
18	Hoch	erledigt	**Administration**	ProductBacklogVorlage	Team	4						4	4
19		laufend	**Meilenstein**	Plan für Benutzerbeobachtung V0	Team								0
20		erledigt	**Meilenstein**	Vision Dokument V0	Team								0
21		Offen	**Meilenstein**	Iteration1 User Stories festgelegt	Team								0
22		Offen	**Meilenstein**	Architekturprototyp Demo, Komponenten & Design entscheide	Team								0
23		erledigt	**Meilenstein**	Review Inhaltsverzeichnis Bericht	Team								0
24		erledigt	**Meilenstein**	Review Draft Wiki Page	Team								0
25		laufend	**Meilenstein**	Review User Stories für nächsten Sprint	Team								0
						Wochentotal	**14.5**	**12.5**	**17**	**18**	**17**	**20**	

Fertig: 16/25

Sprint Beendet

Soll Arbeitstunden Total:	102
Verbleibende Arbeitsstunden:	3
Arbeitsstunden pro Tag:	8.5

Ist Arbeitstunden Total:	99
Ist Arbeitsstunden Amon:	48.5
Ist Arbeitsstunden Patrick:	50.5

Sprint 2

18.10.2010 – 5.11.10

Ziel

4 Gewinnt HTML5 Webseite mit Single Player Modus und Schnittstellenimplementierung

Arbeitsstunden pro Woche

Nr.	Priorität	Status		Aufgabe	Verantwortlich	Soll Arbeitsstunden	Ist Arbeitsstunden						Ist ArbeitstundenTotal
							Woche 1		Woche 2		Woche 3		
1	Hoch	Erledigt	**Scrum Tasks**	Sprint plannen	Team	1	0.5	0.5					1
2	Hoch	Erledigt	**Scrum Tasks**	Sprint review	Team	2					1	1	2
3	Hoch	Erledigt	**Domain Analyse**	Domainanalyse	Team	14	2	1	2	2		3	10
4	Hoch	laufend	**UI Design**	Paperprototyp Version 2	AG	4				2		2	4
5	Mittel	Erledigt	**User Analyse**	Personas	Team	2		2					2
6	Mittel	Erledigt	**User Analyse**	Szenarios	Team	2		2					2
7	Mittel	Erledigt	**User Analyse**	Interview	Team	10		2	4	5		1	12
8	Hoch	Erledigt	**Architektur Design**	Architektur Design	Team	16	2	3	1	2			8
9	Hoch	Erledigt	**Architektur Design**	Technische Evaluation	Team	8	2	1	1	2	5	2	13
10	Hoch	Erledigt	**Implementation**	Apache aufsetzen	PD	3		2		1			3
11	Hoch	Erledigt	**Implementation**	Tomcat aufsetzen	PD	3		2					2
12	Mittel	Erledigt	**User Stories**	User kann einen Zug durchführen von einem PC aus	Team	10			5		5		10
13	Mittel	Erledigt	**User Stories**	User kann den aktuellen Spielstand des Spiels sehen	Team	4					3		3
14	Mittel	Erledigt	**User Stories**	Computer kann Mock-Roboter alle nötigen Befehle senden	Team	12					3	2	5
15	Mittel	Erledigt	**User Stories**	Benutzer sieht eine erste Version der Webseite	Team	2						1	1
16	Mittel	Erledigt	**Administration**	Selbststudium	Team		7	2		3		5	17
17	Mittel	Erledigt	**Administration**	Meetings	Team	12		2	2	2	2	2	10
18		Erledigt	**Meilenstein**	Benutzerbeobachtung beendet	Team								0
19		Erledigt	**Meilenstein**	Vision Dokument V1	Team								0
20		Erledigt	**Meilenstein**	Domain Model	Team								0
21		Erledigt	**Meilenstein**	Review User Stories für nächsten Sprint	Team								0
						Wochentotal	**13.5**	**19.5**	**15**	**19**	**19**	**19**	

Fertig: 20/21

Sprint beendet

Soll Arbeitstunden Total:	105	Ist Arbeitstunden Total:	105
Verbleibende Arbeitsstunden:	0	Ist Arbeitsstunden Amon:	47.5
Arbeitsstunden pro Tag:	8.5	Ist Arbeitsstunden Patrick:	57.5

Sprint 3

8.11.10-26.11.10

Ziel

4 Gewinnt HTML5 Webseite für Telefon optimiert

Arbeitsstunden pro Woche

Nr.	Priorität	Status		Aufgabe	Verantwortlich	Soll Arbeitsstunden	Ist Arbeitsstunden						Ist ArbeitstundenTotal
							Woche 1		Woche 2		Woche 3		
1	Hoch	Erledigt	**Scrum Tasks**	Sprint plannen	Team	1	1	1					2
2	Hoch	Erledigt	**Scrum Tasks**	Sprint review	Team	2					1	1	2
3	Hoch	Erledigt	**Architektur Design**	Design Model	Team	4					3		3
4	Mittel	Offen	**Architektur Design**	Schichtenarchitektur	Team	4					2		2
5	Mittel	Erledigt	**Architektur Design**	Evaluation abschliessen	Team	4	2	3					5
6	Hoch	Erledigt	**UI Design**	UI Design Fertig	Team	4		2		1		1	4
7	Hoch	Erledigt	**Realisierung & Tests**	Code Reviews	Team	6			1				1
8	Hoch	Erledigt	**Realisierung & Tests**	Unit Tests	Team	10			3	3			6
9	Hoch	Erledigt	**Realisierung & Tests**	Algorithmus implementieren	Team	10		5		9		2	16
10	Mittel	Erledigt	**Realisierung & Tests**	Manuelle Tests	Team	4		2		1	3		6
11	Mittel	Erledigt	**Realisierung & Tests**	Usabillity Tests	Team	6			1				1
12	Hoch	Erledigt	**User Stories**	User kann ein Spiel vom Iphone aus spielen	Team	10	1		3	2	5		11
13	Mittel	Offen	**User Stories**	User kann ein Spiel vom Andorid phone aus spielen	Team	5		1				2	3
14	Hoch	Erledigt	**User Stories**	User kann zwischen Spiel gegen Roboter und Spiel gegen anderen Spieler wählen.	Team	15	8		12		2	6	28
15	Niedrig	Erledigt	**User Stories**	User kann ein Spiel vom iPhone aus auf dem Schuhschachtelserver spielen	Team	15	2	5		4		8	19
16	Mittel	Erledigt	**Administration**	Meetings	Team	12	2	2	1	1	2	2	10
17		Erledigt	**Meilenstein**	Domainanalyse fertiggestellt	Team								0
18		Erledigt	**Meilenstein**	UI Design fertiggestellt	Team								0
						Wochentotal	**16**	**21**	**21**	**21**	**18**	**22**	

Fertig: 16/18

Sprint beendet

Soll Arbeitstunden Total:	112	Ist Arbeitstunden Total:	119
Verbleibende Arbeitsstunden:	-7	Ist Arbeitsstunden Amon:	55
Arbeitsstunden pro Tag:	8.5	Ist Arbeitsstunden Patrick:	64

Sprint 4

29.11.10-17.12.10

Ziel

4 Gewinnt Endversion mit Roboter getestet

Arbeitsstunden pro Woche

Nr.	Priorität	Status		Aufgabe	Verantwortlich	Soll Arbeitsstunden	Ist Arbeitsstunden						Ist ArbeitstundenTotal
							Woche 1		Woche 2		Woche 3		
1	Hoch	erledigt	**Scrum Tasks**	Sprint plannen	Team	1	1	1					2
2	Hoch	erledigt	**Scrum Tasks**	Sprint review	Team	2	0.5	0.5					1
3	Mittel	laufend	**Architektur Design**	Design Model	Team	8	3	3	3		2	3	14
4	Mittel	erledigt	**Architektur Design**	Schichtenarchitektur	Team	8	1				1		2
5	Mittel	erledigt	**Architektur Design**	Designentscheidungen dokumentieren	Team	8	1	2	1	2	1		7
6	Mittel	Offen	**Architektur Design**	Matrix App Features	Team	4							0
7	Hoch	erledigt	**Realisierung & Tests**	Code Reviews	Team	16	2	2	2		4		10
8	Hoch	erledigt	**Realisierung & Tests**	Unit Tests	Team	16	2	1	1	2	4	3	13
	Mittel	erledigt	**Realisierung & Tests**	Manuelle Tests	Team	6		2	2	4	2		10
9	Hoch	erledigt	**Realisierung & Tests**	Usabillity Tests	Team	8				4			4
10	Mittel	erledigt	**User Stories**	Während kein Spieler spielt, laufen alte Spiele	Team	8	5		5				10
11	Mittel	erledigt	**User Stories**	User kann sich in Warteliste eintragen	Team	8	2	4		2	4	2	14
9	Hoch	erledigt	**User Stories**	User sieht optimierte Webseite	Team	10	1	6		4	2	16	29
	Mittel	erledigt	**Architektur Design**	Feature Liste		4				4			4
	Mittel	Offen	**Realisierung & Tests**	Benutzerdokumentation anfangen		3							0
12	Mittel	erledigt	**User Stories**	Roboter zeigt alle Spielzüge	Team	6	1		6		14		21
13	Mittel	erledigt	**Administration**	Meetings	Team	6	1	1		1	1	1	5
14		erledigt	**Meilenstein**	Testdokumentation							1		1
15		erledigt	**Meilenstein**	Implementation Beendet									0
16		Offen	**Meilenstein**	Video draft							1		1
					Wochentotal		**20.5**	**22.5**	**20**	**23**	**37**	**25**	

Fertig: 16/20

Sprint beendet

Soll Arbeitstunden Total:	122	Ist Arbeitstunden Total:	148
Verbleibende Arbeitsstunden:	-26	Ist Arbeitsstunden Amon:	77.5
Arbeitsstunden pro Tag:	8.5	Ist Arbeitsstunden Patrick:	70.5

Sprint Abschluss

20.12.10- 23.12.10

Ziel

Abgabe Semesterarbeit

Nr.	Priorität	Status		Aufgabe	Verantwortlich	Soll Arbeitsstu	Ist Arbeitsstunden		Ist ArbeitstundenTotal
							Woche 1		
1	Hoch	erledigt	**Scrum Tasks**	Sprint plannen	Team	1	0.5	0.5	1
2	Hoch	erledigt	**Architektur Design**	Matrix App Features	Team	2	2		2
3	Hoch	erledigt	**Administration**	Video	Team	2	6	1	7
4	Hoch	erledigt	**Administration**	CD	Team	0.5	1		1
5	Hoch	erledigt	**Administration**	Abstract	Team	4		4	4
6	Hoch	erledigt	**Administration**	Poster	Team	4		4	4
7	Hoch	erledigt	**Administration**	Gesamtdokument	Team	4	2	8	10
8	Hoch	erledigt	**Administration**	Korrekturlesen	Team	16	8	14	22
9	Hoch	erledigt	**Architektur Design**	Design Model	Team	8	10		10
10	Hoch	erledigt	**Realisierung & Tests**	Benutzerdokumentation	Team	8	5	4	9
11		erledigt	**Meilenstein**	Projektabgabe	Team				0
						Wochentotal	**34.5**	**35.5**	

Fertig: 11/11

Sprint erledigt

Soll Arbeitstunden Total:	49.5	Ist Arbeitstunden Total:	70
Verbleibende Arbeitsstunden:	-20.5	Ist Arbeitsstunden Amon:	34.5
Arbeitsstunden pro Tag:	8.5	Ist Arbeitsstunden Patrick:	35.5

22 Interview Transkript

Allgemeine Fragen:	
Alle Ihre Aussagen werden Vertraulich behandelt.	
Was ist Ihr Aufgabenbereich?	Ich mache für die ganze Schule alle Veranstaltungen. Die grösste ist der Infotag, welcher jetzt wieder ist. Dieser findet Ende Oktober statt. Es gibt Veranstaltungen, die die Informatik selber organisiert, wie z. B. die Diplomfeier, aber ansonsten machen wir alle Veranstaltugen, wie eben Infotage, Messe- und Schulbesuche. Werbung, Drucksachen
Wie viel Prozent Ihrer Tätigkeit, arbeiten sie für die Informationstage?	Ca. 25 %. Die Organisation ist die Haupttätigkeit, nimmt am meisten Zeit in Anspruch.
Wann braucht es das Objekt?	Messebesuche, Schulebesuche, Infotag an HSR. Informatik zum Anfassen.
Fragen zum Infotag:	
Wie viele Infotage gibt es pro Jahr? (externe und interne)	Es gibt zwei Infotage im Jahr. Nebenbei haben wir während des Winters 22 Schulbesuche (Hauptsächlich Zürich, St.Gallen aber auch z.B. Bern) gemacht. Und dann noch ein paar wenige Messebesuche, d. h. 2 bis 4, wie z. B. an der OBA (Ost-Schweizer Berufsausstellung).
Wie viele Leute kamen an die letzten zwei Infotage?	Beim letzten Infotag waren extrem viele Besucher. Es war um die 450 Besucher. Der langjährige Durchschnitt liegt bei 280 - 300 Besucher. Die letzten zwei Infotage waren wirklich sehr gut besucht. Mal schauen, ob wir diese Zahl nochmals erreichen.
Wie viele Leute sind jeweils gleichzeitig am Stand?	Es kann schon sein das 10 Leute dastehen. Es ist so, etweder kommt einer, dann kommen alle, oder es steht keiner da. Aber eher beim vorbeilaufen. Bei grösseren Ausstellung wie an der OBA gehen wir mit FHO (Fachhochschule Ostschweiz), da sind dann vier Schulen St. Gallen, Buchs, Chur und Rapperswil. Da ist es gut möglich, das es am Stand schon 20 bis 30 Leute sind. Da sind dann auch verschiedene Ausstellungsobjekte nötig.

Was interessiert die Leute am Meisten?	Leute die an den Infotag kommen, sind schon weiter, sie wissen was sie wollen. Sie schauen was die HSR bietet und vergleichen mit anderen Schulen. Die wollen sehr wahrscheinlich studieren, und kommen dem entsprechend sehr gezielt an den Stand und fragen (Was sind die Aufnahmebedingung, reichen meine Kenntnisse). An Messen sprechen wir ein breites Publikum an, welches gar nicht studieren und evtl. auch nie studieren wird. Das ist wie wenn du in einen Laden gehst, ohne das du die Absicht hast etwas zu kaufen. Du schaust dann was es hat. Wenn du dann ein Objekt hast das spannend ist, kannst du die Leute fesseln. Wenn du nur schöne Wände mit Infos und Prospekte hast, kommen nur Leute die Prospekte sammeln. Die Frage ist, wie kannst du Leute abholen, wenn nebendran 100 andere Stände sind. Wenn du ein Objekt hast, das sich bewegt, blinkt oder eine interaktion hast mit diesem kommen die Leute an den Stand. Darum stellt man solche Objekt aus. Wenn sich etwas bewegt und Leute bereits am Stand stehen, kommen sofort mehr Leute. Wenn nur ein Bildschrim mit einer Präsentation da steht, interessiert das niemand. Es ist das spielerische, wenn Leute spielen können, kommen sie.

Gibt es bereits Ausstellungsobjekte, welche am Infotag gezeigt werden? Wenn ja was?	Wir haben fast keine. Ich habe viel zu wenig Objekte. Was wir dabei haben sind diese Eurobot-Roboter, aber diese müssen von Fachleuten betreut werden. Das ist etwas, was die Leute nicht selber bedienen und ausprobieren können. Das ist eher Show. Von den Maschinentechniker gibt es das Balance-Rad. Das ist ein Velorad mit einem Ball obendrauf. Wenn du das siehst, denkst du "easy das kann ich auch" aber schaffst es schliesslich doch nicht. Wenn du es 3 Sekunden schaffst, bist du schon sehr gut. Der Roboter schafft es aber immer. Flickr-Mod, mit man eine Farbe auswählen kann und dann sucht es Bilder mit dem entsprechenden Farbton. Mit Natels die Stichcode lesen können und dann kommt ein Lied. Da haben wir aber Probleme wegen den Lichtverhältnisse. Manchmal funktioniert es nicht richtig, er kann den Strichcode nicht richtig lesen. Scrum-Table hatten wir auch schon an Infotagen, aber auch da haben wir Probleme mit Licht. Wenn das Licht nicht genau da ist wo es sein sollte funktierts nicht richtig. Am besten sind kleinere Objekte die in Autos passen und nicht zu schwer sind, so das man sie alleine Aufstellen kann.
Fragen zum VierGewinnt Roboter:	
Was halten Sie vom Projekt?	Ist gut. Endlich wieder einmal ein neues Ausstellungsobjekt.
Können Sie sich das Aufbauen der Vitrine gemäss Szenario Visiondokument Kap. 9.3 vorstellen?	Ist realistisch. Muss aber transportable sein. Darf nicht zu gross sein. Es sollte in den HSR-Bus (Opel Bus) passen. Gut wäre, wenn es aus mehreren Teilen bestehen würde, so das man es alleine transportieren kann. Z. B. der unter Teil mit der Technik (Server, Verkabelung etc.) und dann der obere Teil mit dem Roboter und optional noch eine Glasvitrine zum draufsetzen.

Werden Sie die Vitrine Aufbauen? Wie gut sind Ihre Computerkenntnisse? Wie gut sind die Computerkenntnisse der Person, die die Vitrine aufstellen wird?	Also man kann es so sagen, die Leute, welche beim Stand stehen sind Standpersonal. Ich gehe den Stand auf- und abbauen, aber danach bin ich nicht den ganzen Tag dort. Sondern das Personal unserer Verwaltung. Diese Leute haben sehr unterschiedliche Computerkenntnisse. Von keine Ahnung bis sehr gute Kenntnisse von Computern. Es muss auch jemand die Vitrine aufbauen und betreiben können, ohne grosse Computerkenntnisse zu haben. Mit einer einfachen Anleitung können es zwei bis drei Klicks sein, die die Leute machen müssen. Aber nicht "gehen Sie in die Netzwerkeinstellung, dann Eigenschaften, machen Sie das....". Möglichst die Vitrine an den Strom anschliessen, das ganze starten, zwei bis drei Klicks und dann läufts. Wie z. B. das Flickr-Modus, dieses Ausstellungsobjekt kannst man einstecken, Computer aufstarten und dann auf dem Desktop das Symbol anklicken. Fertig. So ist das super.
Wie lange darf der Aufbau maximal dauern?	10 - 15 Minuten. Aber je kürzer umso besser. Wenn der Aufbau lange dauert, wird das Ausstellungsobjekt eher auf Messen mitgenommen, welche zwei bis drei Tage dauern. Wenn wir auf einen Schulbesuch gehen, welcher nur eine Stunde dauert, wird es nicht mitgenommen. Aber wenn es schnell geht, das Objekt aufzustellen, sagen wir 5 Minuten, dann nimmt man es mit. Wenns 15 Minuten geht, dann überlegt mans sich schon, ob man es mitnimmt.
Gibt es bei den diversen Ausstellung eine Internetanbindung?	In Schulen gibt es meistens eine Internetverbindung die verwendet werden kann. Allerdings auf den Messen nicht, oder nur selten. Dort muss man selber eine Lösung finden.
Was halten Sie von den Leistungsanforderungen, die für das Projekt definiert wurden? (Visiondokument Kap. 5)	Sehr gut.
Wie schätzen Sie die Computer-, im speziellen die Mobile-Phone-Kenntnisse der Besucher ein?	Sehr unterschiedlich, aber es sollte nicht zu schwierig sein, sich mit dem Server zu verbinden, weil wenn es bei den Besucher nicht gleich funktioniert, sind sie frustriert und gehen weg. Die meisten versuchen nicht lange sich zu verbinden

Sehen Sie sich bitte den ersten Prototyp der Benutzeroberfläche für das iPhone an. Was halten Sie davon? Was würden Sie ändern?	Wo muss geklickt werden, wenn man einen Stein setzen will? Muss ein Zug nochmals bestätigt werden? Was passiert wenn geklickt wurde, während der Roboter arbeitet? Sieht man hier die Steine, die gesetzt wurden? Man muss ja gar nicht mit dem Roboter spielen, da du ja das Spielfeld direkt auf dem Mobile Phone hast. Was ist, wenn es eine Internetverbindung zum Roboter gibt, und ein Besucher die Internetadresse kennt. Kann er diese dann Zuhause in den Browser gehen und spielen? Wenn man zu zweit spielen kann ist es sicherlich lustiger. Wenn nur jemand spielen kann ist es relativ langweilig. Mit mehrer Leuten die Spielen hat man gleich mehr Bewegung beim Stand.
Welche funktionen haben Ihrer Meinung nach Priorität, um den grössten Werbeeffekt zu erzielen?	Mehrere Leute wäre super. Ein Spieler ist sicher schon spannend, aber mehrer wäre einfach besser. Mehrfachspieler, demokratisches System oder alle gegen den Computer, wäre das Tüpfchen auf den I. Zu zweit spelen ist schon super, aber Mehrspielemodus wäre noch eine Steigerung.
Wie sieht der Roboter der HTW Chur aus? Was sind dessen Funktionen?	Ich habe den Roboter leider noch nie Live gesehen nur auf einem Bild. Aber dieser Roboter ist schon ein älteres Projekt. Es ist ein Brett mit einem Roboter drauf, der die Steine drauf tut. Die Bedienung erfolgt über Knöpfe an dem Objekt. Ich habe mit den Leuten von Chur telefoniert und ihnen mitgeteilt, das wir auch ein ähnliches Projekt machen. Die sagten es wäre überhaupt keine Problem. Bei uns ist das Hauptaugenmerk auf der Bedienung mit dem Mobiltelefon und was man mit der heutigen Technik alles machen kann. Das Spiel ist eher der Nebeneffekt. Das von Chur ist eher rudimentär. Das man zu zweit spielen kann unterscheidet unseres Projekt von dem von Chur.
Sonstige Ideen und Inputs?	Es ist eine super Idee. Gut wenn wir wieder einmal etwas neues haben um den Besucher zu zeigen. Vieleicht am Objekt eine Datencard, mit welcher sich der Server an den Webserver bei der HSR verbinden kann, z. B. Natel. Infotag am 30.10. von 9.45 - 16.00 Uhr. Informatikaustellung von 11.30 bis 14.00 Uhr

23 Erklärung über die eigenständige Arbeit

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.

Ort, Datum — Amon Grünbaum

Rapperswil, 23.12.2010

Ort, Datum — Patrick Dünser

Rapperswil, 23.12.2010