



Pflichtenheft

Projekt: Harambe Wins

Team: Harambe

Teammitglieder: Jann Wickert, Robert Schulz, Krishna Parthiparajan, Michael Schlechtinger, Peter Ulbrich, Tim Wagner

Zeitraum: 25.09. – 14.11.2016



Änderungshistorie

Datum	Version	Status	Änderungen
25.09.2016	0.1	Entwurf	Erstellung des Inhalts und Kapitel 1
03.10.2016	0.2	Entwurf	Kapitel 2 und 3 verfasst, optische Anpassungen
10.10.2016	0.3	Entwurf	Kapitel 4 und 5 verfasst
17.10.2016	0.4	Entwurf	Alle Kapitel überarbeitet
12.11.2016	0.5	Entwurf	Testings und restliche Kapitel hinzugefügt
14.11.2016	1.0	Verabschiedet	Alle Kapitel überarbeitet



Contents

1	Zielbestimmung.....	1
1.1	Musskriterien.....	1
1.2	Wunschkriterien	3
1.3	Abgrenzungskriterien	3
2	Produkteinsatz	3
2.1	Anwendungsbereich	3
2.2	Zielgruppen.....	3
2.3	Betriebsbedingungen.....	3
3	Produktübersicht	4
3.1	Technische Sicht	4
3.2	Styleguide	4
4	Produktfunktionen.....	7
4.1	Anwendungsfälle	7
5	Produktdaten	9
6	Produktleistungen.....	10
7	Qualitätsanforderungen	11
8	Testszenarien und Testfälle	12
9	Nichtfunktionale Anforderungen.....	13
10	Technische Produktumgebung.....	13
10.1	Software.....	13
10.2	Hardware	13
10.3	Orgware	14
11	Ergänzungen.....	14
12	Glossar	14

1. Zielbestimmung

Es soll ein Software-Agent gebaut werden, der autonom das Spiel „Vier-Gewinnt“ gegen einen anderen Spieler spielen kann.

1.1 Musskriterien

Der Agent

- ◇ muss mit dem vorgegebenen Server kommunizieren können
- ◇ muss entsprechend der Importanleitung für Dozenten zum Import möglich sein
- ◇ muss auf dem Studentenrechner lauffähig sein
- ◇ soll keine Compilerfehler nach dem Import in Eclipse anzeigen
- ◇ Fehlerfreier Ablauf des zentralen Geschäftsprozesses

Anforderungen an die Benutzerschnittstelle:

- ◇ Grafische Oberfläche
- ◇ Anzeige des Satzstatus
- ◇ Anzeige des Spielstandes
- ◇ Anzeige des Spielfeldes
- ◇ Anzeige aller Züge des aktuellen Satzes auf dem Spielfeld
- ◇ Realisierung auf Basis von JavaFx oder Swing

Schnittstellen

- ◇ Agent und Server kommunizieren zu einem Zeitpunkt über eine von zwei Schnittstellen, d. h. über die Datei-Schnittstelle oder die Pusher-Schnittstelle
- ◇ Während eines Satzes erfolgt kein Wechsel der Schnittstellen. Zwischen den Sätzen, d. h. also auch während eines Spiels, kann die Schnittstelle gewechselt werden. Der Wechsel der Schnittstelle wird angesagt

Protokoll

- ◇ Alle Mitteilungen erfolgen im jeweils gleichen Ablaufkontext und zu jeweils gleichen zeitlichen Bedingungen.
- ◇ Kommunikationsschicht: Aus technischer Sicht unterscheiden sich die Schnittstellen wesentlich, vor allem hinsichtlich der verwendeten Kommunikationsdienste und -primitive (Datei vs. Event)

Datei-Schnittstelle

- ◇ Kommunikation von Server und Agents über Dateien auf der Basis von Streams

Pusher-Schnittstelle

- ◇ Kommunikation von Server und Agents über Events auf der Basis von WebSockets
- ◇ Es wird ein vom Pusher-Anbieter gehosteter WebSocket-Server verwendet
- ◇ Credentials müssen am Agenten einstellbar sein
- ◇ Authentifizierungsdaten zum Tunier werden vorgegeben
- ◇ Der Agent kommuniziert mit dem Vier-Gewinnt-Server über genau eine Pusher-App bidirektional auf genau einem Pusher
- ◇ Der Agent hat genau eine distinkte Pusher-App

Datenhaltung

- ◇ Speichern der Daten in einer Datenbank
- ◇ automatische Speicherung der Daten durch den Agenten
- ◇ zu speichernde Daten sind:
 - Gegner, Startspieler, Sieger, Punkte, Spiele, Sätze, Züge
 - minimal: alle Daten nach Maßgabe des Punktemodus
- ◇ Realisierung auf der Basis von HSQLDB
- ◇ wenigstens drei Abfragen bzw. Abfragevarianten implementieren;
Minimalanforderung:
 - alle gespielten Spiele anzeigen
 - alle Züge eines frei wählbaren Spiels der angezeigten Spiele anzeigen
- ◇ wenigstens drei Spiele für Beispielabfragen mit Release-Version ausliefern

Dokumentation

- ◇ Anfertigen einer Entwicklungsdokumentation
- ◇ Anfertigen einer Benutzungsdokumentation

Sonstiges

- ◇ Es dürfen nur die vom Dozenten zugelassenen Materialien (Tools, Dokumentation, Bibliotheken, Templates usw.) verwendet werden

1.2 Wunschkriterien

- ◇ Der Agent darf wesentlich mehr leisten, sofern dies Ablauf und Kommunikation nicht beeinträchtigt
- ◇ Es können weitere Bestandteile zur Dokumentation hinzugefügt werden

1.3 Abgrenzungskriterien

- ◇ Das Spiel soll vorerst in der deutschen und englischen Sprache veröffentlicht werden
- ◇ Das Spiel soll für den Spaß- und Lernfaktor erstellt werden und keine finanziellen Hintergründe beinhalten

2. Produkteinsatz

2.1 Anwendungsbereich

Einzelpersonen verwenden diesen Dienst zum Spielen des Spiels „Vier-Gewinnt“. Diese Plattform soll dem Einzelnen eine Kommunikation mit Gleichgesinnten ermöglichen.

2.2 Zielgruppen

Die Zielgruppe sind Personengruppen, die kurz zur Ablenkung z.B. in der Mittagspause, gerne an Gesellschaftsspielen teilhaben möchten. Diese Plattform ist für Einzelpersonen gedacht, die eine kurze Ablenkung suchen. Vor allem die jüngere Generation soll durch das Thema und den Aufbau der Spielgestaltung angesprochen werden.

2.3 Betriebsbedingungen

Das System soll bezüglich der Betriebsbedingungen nicht wesentlich von anderen Desktopdiensten bzw. -anwendungen unterscheiden.

3. Produktübersicht

3.1 Technische Sicht

Das Produkt Harambe Wins ist auf einem multi-layer Design aufgebaut, welches folgende Schichten beinhaltet:

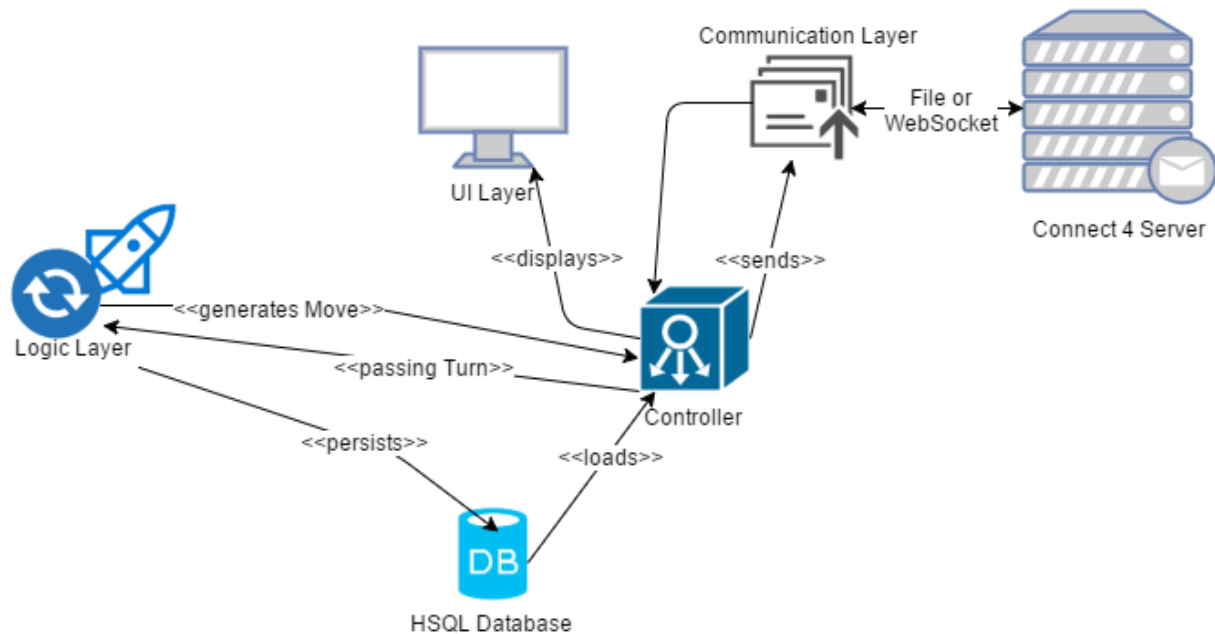


Abbildung 1: Schichtenmodell von Harambe Wins; technische Abbildung

Die Kommunikationsschicht dient als Interface zum Vier-Gewinnt Server und ermöglicht somit die Kommunikation zum File- als auch zum Pusher WebSocket Interface. Diese Schicht wird vom main UI Thread getrennt, damit die Benutzer mit dem Interface agieren können während man auf die Antworten des Servers wartet.

Die Logic Layer beinhaltet die Funktionalitäten für die künstliche Intelligenz. Als Grundlage wird hierfür der Minimax-Suchalgorithmus mit Alpha-Beta-Pruning verwendet.

3.2 Styleguide

Das Produkt soll ansprechend für die jüngere Generation gebaut sein. Ein wichtiger Bestandteil ist Harambe und die Dschungel-Atmosphäre. Das Logo des Spiels soll mit berühmten Konsolenspielen assoziiert werden. Deshalb soll das Logo rundlich mit einem Banner erstellt werden, der den Namen des Spiels trägt. Um dieses tropische Motiv umzusetzen soll das Design mit Elementen der Natur (z.B. Holz und Pflanzen) gestaltet werden. Das Dschungel-Logo soll vor allem im Vier-Gewinnt Spielfeld vorhanden sein.

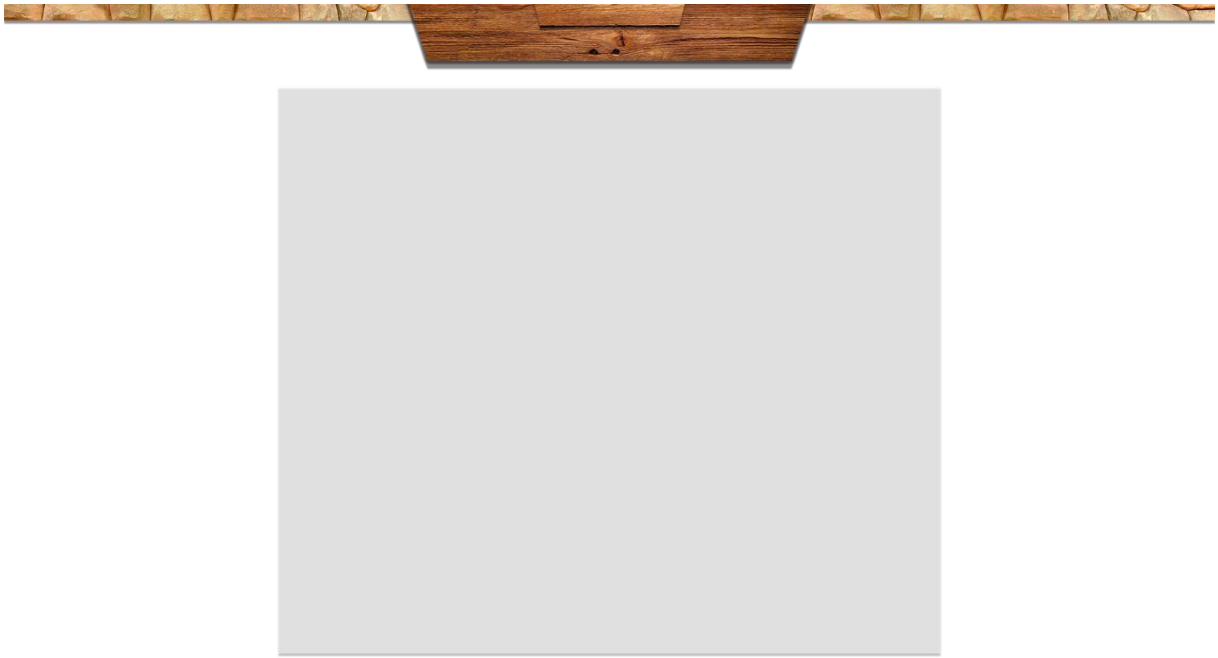


Abbildung 1: Spielständeübersicht des Spiels

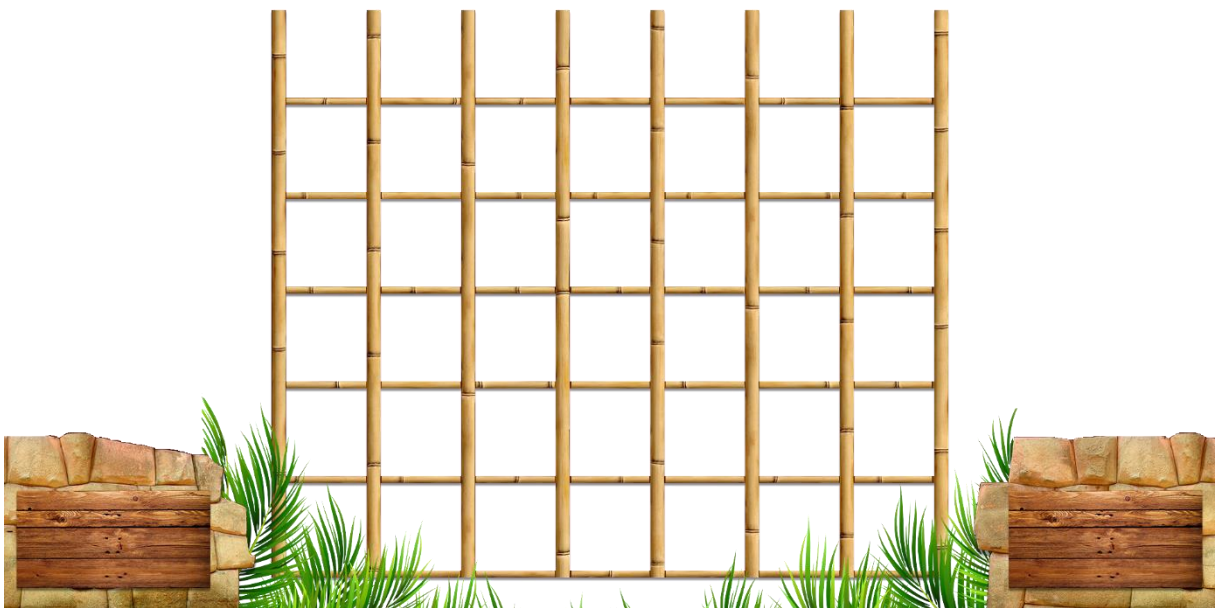


Abbildung 2: Spielbrett für das Vier-Gewinnt Spiel



Abbildung 3: Startseite von Harambe Wins

Der Hintergrund muss nicht einer tropischen Atmosphäre entsprechen, sondern wird zufällig jedes Spiel angepasst. Der Hintergrund soll mit Animationen gestaltet werden.

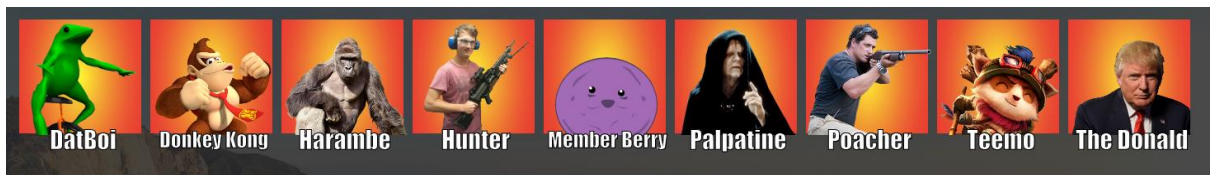


Abbildung 4: Charakterauswahl

Folgende Charaktere sind in dem Spiel vertreten:

DatBoi	Der Einrad-fahrende Frosch erlangte seine Bekanntheit durch das Internet.
Donkey Kong	Der wohl bekannteste Gorilla der Welt. Diese Spielikone stellt einen perfekten Konkurrenten für Harambe dar.
Harambe	Der Star dieses Spiels ist Harambe. Nach seiner Erschießung im Zoo von Cincinnati wurde er zur Berühmtheit.
Hunter	Ein menschlicher Widersacher für Harambe.
Member Berry	Der Serie South Park verdankt Member Berry ihre Berühmtheit.
Palpatine	Die Star Wars Legende kehrt zurück.
Poacher	Ein weiterer menschlicher Widersacher für Harambe.
Teemo	Der Teufel von League of Legends
The Donald	Der designierte Präsident der vereinigten Staaten.

4. Produktfunktionen

4.1 Anwendungsfälle

Unter Anwendungsfall, auch use case genannt, versteht man einen typischen und wesentlichen Vorgang vom Nutzer bei der Programmbedienung. Ein Anwendungsfall ist in diesem Kontext eine Sequenz von Transaktionen, die von einem Akteur im Dialog mit dem System ausgeführt werden, um für den Akteur einen messbaren Nutzen zu erzielen. Transaktion ist hier (datenbank-) technisch zu verstehen, insbesondere werden Transaktionen ganz oder gar nicht wirksam.

/LF 10/

Anwendungsfall: Spiel starten

Ziel: Das Spiel soll geöffnet werden

Kategorie: primär

Vorbedingungen: -

Nachbedingungen Erfolg: Öffnen des Startbildschirms

Nachbedingungen Fehlschlag: Der Startbildschirm öffnet sich nicht

Auslösendes Ereignis: -

Beschreibung:

1. Das Spiel wird mittels einer JAR-Datei geöffnet
2. Erscheinen des Startbildschirms

Erweiterung: -

Alternativ: -

/LF 20/

Anwendungsfall: Hauptmenü aufrufen

Ziel: Das Hauptmenü mit den Hauptfunktionalitäten soll geöffnet werden

Kategorie: primär

Vorbedingungen: Die Startseite wurde geöffnet

Nachbedingungen Erfolg: Öffnen des Hauptmenüs

Nachbedingungen Fehlschlag: Der Hauptmenü öffnet sich nicht

Auslösendes Ereignis: -

Beschreibung:

1. Weiterleitung von der Startseite auf die Hauptmenüseite
2. Auswahl der Hauptfunktionalitäten ist möglich

Erweiterung: -

Alternativ: -

/LF 30/

Anwendungsfall: Spiel im Online-Modus ausführen

Ziel: Das Spiel soll im Online-Modus ausgeführt werden

Kategorie: primär

Vorbedingungen: -

Nachbedingungen Erfolg: Öffnen des Startbildschirms

Nachbedingungen Fehlschlag: Der Startbildschirm öffnet sich nicht

Auslösendes Ereignis: -

Beschreibung:

1. Auswahl des Online-Modus

Erweiterung: -

Alternativ: Spiel wird im Offline-Modus ausgeführt

/LF 40/

Anwendungsfall: Spiel im Offline-Modus ausführen

Ziel: Das Spiel soll im Offline-Modus ausgeführt werden

Kategorie: primär

Vorbedingungen: -

Nachbedingungen Erfolg: Öffnen des Startbildschirms

Nachbedingungen Fehlschlag: Der Startbildschirm öffnet sich nicht

Auslösendes Ereignis: -

Beschreibung: Auswahl des Offline-Modus

Erweiterung: -

Alternativ: Spiel wird im Online-Modus ausgeführt

/LF 50/

Anwendungsfall: Daten in Datenbank speichern

Ziel: Der Spielverlauf soll in der Datenbank gespeichert werden

Kategorie: primär

Vorbedingungen: -

Nachbedingungen Erfolg: Anzeige von den Spielzügen ist möglich

Nachbedingungen Fehlschlag: Die Spielzüge werden im Verlauf nicht angezeigt

Auslösendes Ereignis: Durchführen eines Spielzugs

Beschreibung:

1. Ein Spielzug wird durchgeführt
2. Speichern in der Datenbank

Erweiterung: -

Alternativ: -

/LF 60/

Anwendungsfall: Anzeigen eines Spielverlaufs

Ziel: Die durchgeführten Spielzüge sollen angezeigt werden

Kategorie: primär

Vorbedingungen: Ein Spiel muss zu Ende gespielt werden

Nachbedingungen Erfolg: Anzeige von den Spielzügen ist möglich

Nachbedingungen Fehlschlag: Die Spielzüge werden im Verlauf nicht angezeigt

Auslösendes Ereignis: Durchführen eines Spielzugs

Beschreibung:

1. Das Spiel oder ein Satz wird beendet
2. Umschalten auf das Hauptmenü
3. Die Spielzüge können angezeigt durch den Replaybutton angezeigt werden

Erweiterung: -

Alternativ: -

5. Produktdaten

In diesem Bereich wird angegeben, welche der Daten langfristig in der Datenbank gespeichert werden.

Jeder Punkt **/DXXX/** stellt im Prinzip einen Datensatz dar. (PK) steht für Primärschlüssel.

/D010/ Game: Alle Informationen zu einem Spiel:

- ◇ Game_uuid (PK)
- ◇ our_player
- ◇ opponent_player
- ◇ our_points
- ◇ opponents_points
- ◇ we_won
- ◇ time

/D020/ Set

- ◇ game_uuid (PK)
- ◇ set_number (PK)
- ◇ we_started
- ◇ we_won

/D030/ Turn: Alle Informationen zu einem Spielzug

- ◇ Game_uuid (PK)
- ◇ set_number (PK)
- ◇ turn_number (PK)
- ◇ is_opponent
- ◇ column

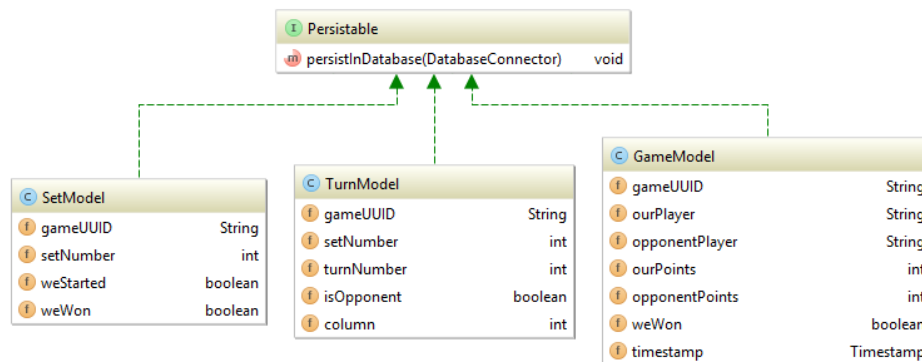


Abbildung 2: UML-Klassendiagramm des Datenbankmodells

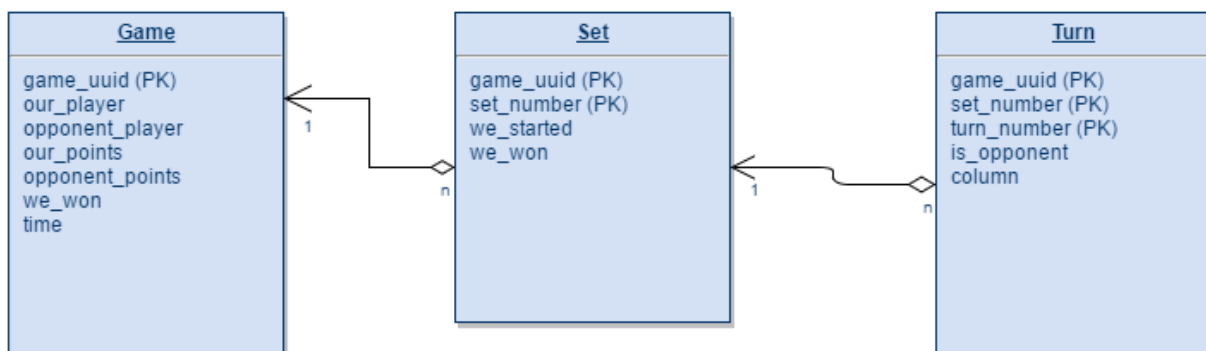


Abbildung 1: Entity-Relationship-Diagramm des Datenbankmodells

6. Produktleistungen

Im Rahmen dieses Projektes gibt es drei verschiedene Leistungen, die erbracht werden müssen.

Die erste Version der Leistung entspricht einem Prototyp, der zu Beginn der vierten Veranstaltung abgegeben werden muss. Bei der Abgabe wird geprüft, ob der Import gemäß der im Glossar angehängten Importanleitung für den Dozenten möglich ist. Nach dem Import in Eclipse dürfen keine Compilerfehler angezeigt werden. Laufzeitfehler auf dem Dozentenrechner sind allerdings zulässig. Es ist ein auf dem Studentenrechner lauffähiger und

vorzeigbarer Oberflächenprototyp gefragt. Optional ist die Demonstration weiterer Funktionalitäten.

Die zweite Version der Leistung entspricht einer Beta, die zu Beginn der sechsten Veranstaltung erbracht werden muss. Sie muss die Anforderungen des Prototyps erfüllen und zusätzlich mindestens fünf Züge über den Server spielen können. Beide Schnittstellen müssen funktionieren. Ein Zufallszug des eigenen Agenten genügt. Eine Siegmustererkennung oder Spalte-ist-voll-Erkennung ist nicht notwendig.

Die finale Abgabe entspricht dem Release, die am Ende der letzten Veranstaltung abgegeben werden muss. Es soll ein auslieferungsfähiges Bundle, welches die Dokumentation erhält und dem Quellcode als Menge von Dateien auf CD oder DVD abgegeben wird. Auf Basis der Dokumentation soll der Anwender den Agenten installieren, konfigurieren und bedienen können. Ein Entwickler sollte in der Lage sein das Programm zu warten und zu erweitern.

7. Qualitätsanforderungen

Bei der Softwarequalitätsanforderungen werden bestimmte Punkte der Norm ISO/IEC 9126 betrachtet.

Produktqualität	Sehr gut	Gut	Normal	Nicht relevant
Funktionalität				
Angemessenheit		X		
Richtigkeit		X		
Interoperabilität	X			
Ordnungsmäßigkeit		X		
Sicherheit			X	
Zuverlässigkeit				
Reife	X			
Fehlertoleranz	X			
Wiederherstellbarkeit				X
Benutzbarkeit				
Verständlichkeit	X			
Erlernbarkeit	X			
Bedienbarkeit	X			
Effizienz				
Zeitverhalten	X			
Verbrauchsverhalten	X			

8. Testszenarien und Testfälle

Für das Projekt wurde die kontinuierliche Integration als Testfall verwendet.

„Continuous integration (CI) is a software engineering practice in which isolated changes are immediately tested and reported on when they are added to a larger code base. The goal of CI is to provide rapid feedback so that if a defect is introduced into the code base, it can be identified and corrected as soon as possible. Continuous integration software tools can be used to automate the testing and build a document trail.“¹

Aus diesem Grund benutzt unser Projekt Travis CI, welches automatisch alle Unit Tests in das Repository nach jedem Push und Merge ausführt, um mögliche Konflikte zu vermeiden.

Default Branch					
✓ master 1 builds	→ #1 passed 15 days ago	37f70e3 Ulrich	✓		
Active Branches					
✓ feature/database 8 builds	→ #65 passed about 2 hours ago	755ef5d PeterUlrich	✓	✓	✓
✓ feature/pusherinterface 4 builds	→ #61 passed about 10 hours ago	fd8fa0 Jann Wickert	✓	✓	✓
✓ develop 16 builds	→ #59 passed about 11 hours ago	dc598d0 mschlechtinger	✓	✓	✓
✓ feature/board_ui_finish 14 builds	→ #58 passed about 11 hours ago	0ffa8ce mschlechtinger	✓	✓	✓
✓ feature/documentation 4 builds	→ #47 passed a day ago	73b3a23 PeterUlrich	✓	✓	✓
✓ feature/algorithm 8 builds	→ #36 passed 7 days ago	b651cb1 PeterUlrich	✓	✓	✓
✓ feature/start_menu 1 builds	→ #35 passed 7 days ago	ef4f645 WagT	✓		
✓ feature/ui_test 9 builds	→ #20 passed 10 days ago	110f7a3 mschlechtinger	✓	✓	✓

Abbildung 5: Gesamtübersicht von Travis CI in unserem Repository

Default Branch					
✓ master 1 builds	→ #1 passed 15 days ago	37f70e3 Ulrich	✓		

Abbildung 6: Einzelsicht Unit Test

Ein Unit Test, oder auch Modultest genannt, wird angewandt um funktionale Einzelteile einer Software zu testen. Dabei soll die Funktionalität dieses Einzelteils geprüft werden.

¹ Zitat von SearchSoftwareQuality; <http://searchsoftwarequality.techtarget.com/definition/continuous-integration>

9. Nichtfunktionale Anforderungen

- ◇ Während der gesamten Projektlaufzeit ist die Bereitstellung eines Zugangs zum Teamrepository für den Dozenten erforderlich
 - Form: eine URL per Mail an den Dozenten
 - im Teamrepository sollen alle entwickelten Artefakte, insbesondere der aktuelle Stand des Quellcodes und der Dokumentation, zugreifbar sein
 - Nur Arbeitsergebnisse, die für den Dozenten im Repository zugreifbar sind, werden berücksichtigt
- ◇ Die dem Kunden (dem Dozenten) zur Verfügung gestellten Artefakte sollen mit allgemeinen Standardprogrammen auf einem normalen MS-Windows-PC angesehen bzw. verwendet werden können. In diesem Sinne geeignete Formate und Werkzeuge sind:
 - *.java, *.txt, *.jpg, *.bmp, *.pdf, *.html, *.css, *.jar, *.zip
 - Open-Office, MS-Office
 - kein Visio, MS-Projekt, Flash, Silverlight, QuickTime o. ä.
 - keine (anderen) speziellen Plugins oder Clients beim Kunden (Dozenten) voraussetzen.
- ◇ Die Verwendung spezifischer Vier-Gewinnt-Algorithmen, -Codes und -Dokumentation aus anderen Quellen dürfen nicht verwendet werden

10. Technische Produktumgebung

Welche Software, Hardware und Orgware wird zur Entwicklung benötigt?

10.1 Software

- ◇ Betriebssystem: Win 7 Enterprise SP1, 32 Bit
- ◇ Tools: Eclipse Neon
- ◇ Laufzeitumgebung: JRE 1.8_101

10.2 Hardware

Client

- ◇ Internetfähiger Rechner
- ◇ PC: Intel Xeon processor E3-1245, 64 Bit

Server

- ◇ Internetfähiger Server
- ◇ Rechner, der die Ansprüche der Server-Software erfüllt
- ◇ Ausreichend Rechen- und Festplattenkapazität

10.3 Orgware

- ◇ Technologie:
 - J2SE (nach Maßgabe der Version im jeweiligen PC-Raum)
 - JavaFx
 - HSQLDB

11. Ergänzungen

Die Abgabe des Produkts soll auf einer CD/DVD samt Dokumentation und gespeichert und abgegeben werden.

Des Weiteren wurden mehrere Dokumentationen für Harambe Wins angelegt:

Link	Beschreibung
Harambe Wiki	Hier geht es zum Wiki, um weitere Angaben zur Projektentwicklung und Technologie zu finden.
JavaDocs	Die vollständige JavaDoc zu Harambe Wins

12. Glossar

- Unit Test** Ein Unit Test, oder auch Modultest genannt, wird angewandt um funktionale Einzelteile einer Software zu testen. Dabei soll die Funktionalität dieses Einzelteils geprüft werden.
- Dokumentation** In diesem Fall wird eine Produktdokumentation gemeint. Diese Produktdokumentation beinhaltet eine Entwicklungsdokumentation und eine Benutzungsdokumentation.