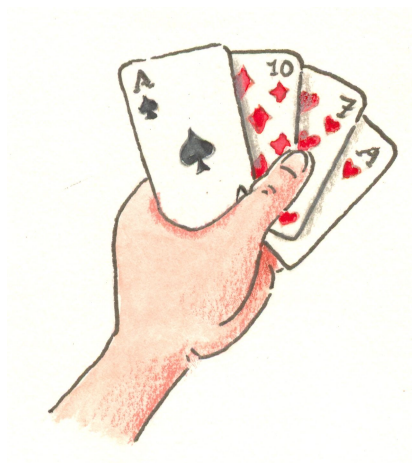


Validierungsbericht

19. Januar 2014



NET-WizHearts

Phase	Verantwortlicher	E-Mail
Pflichtenheft	Alina Meixl	alina@meixl.de
Entwurf	Viktoria Witka	witkaviktoria@freenet.de
Spezifikation	Daniel Riedl	dariedl14@yahoo.de
Implementation	Andreas Altenbuchner	a.andi007@gmail.com
Validierung	Patrick Kubin	kubin@fim.uni-passau.de
Präsentation	w	w

Inhaltsverzeichnis

1	Einleitung	2
2	Änderungen am Spiel	2
2.1	Änderungen zu vorherigen Phasen	2
2.2	Bereits bekannte und korrigierte Bugs	2
2.3	Während der Validierung gefundene Bugs	3
3	Plichtenheft Tests	4
4	Line- und Branch-Coverage	8
4.1	View	8
4.1.1	Interaktiver Test	8
4.1.2	JUnitTest-Ergebnisse	9
4.2	Client	11
4.3	Server	12
4.4	Ruleset	13
4.5	ComObjects	13
4.6	Alle Pakete und Klassen	14
5	Korrektheitstests	15
6	Skalierungs- und Belastungs-Tests	15
6.1	Lobby	15
6.1.1	Werte	15
6.1.2	Zusammenfassung	17
6.2	Server	18
6.2.1	Werte	18
6.2.2	Zusammenfassung	21
7	Weitere Tests	21
7.1	BETA-Test	21
8	Fakten zum Programm NETWizHearts	22

1 Einleitung

Im folgenden Dokument werden alle Tests und ihre Ergebnisse aufgeführt, die im Rahmen der Testphase des Spiels NET-WizHearts abgeschlossen wurden, um die Funktionalitäten des Programms zu testen und deren Korrektheit zu bestätigen.

2 Änderungen am Spiel

2.1 Änderungen zu vorherigen Phasen

- **Kartenanzeige/Timer:** Um das Spiel benutzerfreundlicher zu gestalten, wurde ein Timer eingefügt, der dafür sorgt, dass, wenn der letzte Spieler eines Umlaufes seine Karte ablegt, alle ausgespielten Karten noch etwa eine Sekunde angezeigt werden, damit auch andere Mitspieler die zuletzt gelegte Karte erkennen können und sehen, wer den Stich gemacht hat.
- **Sortierung:** Die Karten der beiden Spiele werden nun sortiert, um das Spiel übersichtlicher zu gestalten.
- **Spielende:** Es wurde eine Umfrage hinzugefügt, die es den Spielern erlaubt, nach dem Spielende direkt ein neues Spiel zu starten, in der selben Zusammensetzung wie zuvor, sofern jeder Spieler zustimmt.
- **Buttons:** Buttons, die nicht gedrückt werden können sollen, sind nun ausgegraut, statt keine Reaktion hervorzurufen.
- **Rundenanzeige:** In dem Bereich, in welchem zuletzt die eigenen Informationen des Spieler angezeigt wurden, wird nun auch noch die Rundenzahl angezeigt.

2.2 Bereits bekannte und korrigierte Bugs

Die Einstufung der Bugs erfolgt nach folgenden drei Kriterien:

- **kritisch:**
Der Fehler muss dringend behoben werden, da die Funktionen durch den Fehler stark beeinträchtigt werden.
- **wichtig:**
Der Fehler sollte behoben werden, da Funktionen nicht einwandfrei genutzt werden können oder Verwirrungen für den Benutzer entstehen.
- **wünschenswert:**
Es wäre gut, wenn der Fehler behoben würde. Da es sich jedoch um einen Schönheitsfehler handelt, ist dies nicht zwingend nötig, da die Funktionen des Programms nicht weiter eingeschränkt werden.

Es folgen aus der Implementierungsphase bekannte Bugs mit Einstufung und Ergebnis:

- **Hearts - Spielen der Karten:** Es wird beim Spielen von Hearts vom ServerRuleset keine Nachricht an alle ClientRulesets gesendet, wenn Herz gebrochen wird. Dadurch entsteht der Fehler, dass nur derjenige, der Herz gebrochen hat, davon weiß und deshalb die anderen Mitspieler dennoch kein Herz spielen können.
Einstufung: kritisch, da das Regelwerk nicht fehlerfrei funktioniert. Dies macht Spielen unmöglich.
Ergebnis: Der Fehler wurde behoben, die benötigte Nachricht wird jetzt gesendet. Dazu wurde eine neue RulesetMessage 'MsgBoolean' angelegt.
- **Abgelegte Karten:** Wird bei einem Umlauf im Spiel die letzte Karte gespielt, verschwinden alle abgelegten Karten in diesem Moment. Die anderen Spieler haben so keine Möglichkeit, die zuletzt gespielte Karte zu erkennen. Dadurch ist es kaum möglich den Spielverlauf nachzuvollziehen. Deshalb sollen die Karten noch einige Sekunden angezeigt werden, bevor ein neuer Umlauf beginnt.
Einstufung: wichtig, da es unter Umständen nötig ist, die gespielten gegnerischen Karten sehen zu können, um eine Strategie zu entwickeln.
Ergebnis: Der Fehler wurde behoben. Die Karten werden nun 1,2 Sekunden angezeigt.

- **Heartspunkte:** Beim Spiel Hearts wird bei den Informationen von jedem Spieler die Anzahl der gemachten Stiche statt der Punkte angezeigt.
Einstufung: wichtig, da Informationen über die Punkte der gegnerischen Spieler ein strategisches Spielen ermöglichen, außerdem ist die Anzeige der eigenen Punkte von Nöten, um einen Spielüberblick zu behalten.
Ergebnis: Der Fehler wurde behoben, die Anzeige enthält jetzt die richtigen Informationen.
- **GUI Hearts:** Im Spiel Hearts werden rechts unten im Spielfenster Komponenten angezeigt, die das Spiel nicht benötigt, beispielsweise die Anzeige einer Trumpfkarte.
Einstufung: wünschenswert, da ein Spieler womöglich durch funktionslose Felder auf dem Spielfeld verwirrt werden könnte, der Spielablauf wird dadurch allerdings zu keinem Zeitpunkt eingeschränkt.
Ergebnis: Der Fehler wurde behoben. Unnötige Komponenten sind jetzt nicht mehr sichtbar.
- **Fehlermeldungen:** Einige Fehlermeldungen können noch optimiert werden. So enthält beispielsweise die englische Fehlermeldung bei der Eingabe eines falschen Passwortes einen Rechtschreibfehler im Wort 'incorrect'.
Einstufung: wünschenswert, da alle Fehlermeldungen korrekt sind, jedoch Ausdrucks- und Rechtschreibfehler enthalten.
Ergebnis: Der Fehler wurde behoben. Die Fehlermeldungen sind optimiert.
- **Kartenrückseite unter Linux:** Die Bilder der Kartenrückseiten werden unter Linux nicht sofort initialisiert, wenn das Spielfenster geöffnet wird, sondern erst beim Spielstart.
Einstufung: wünschenswert, da es möglich ist das Spiel fehlerfrei zu spielen, da nur die Kartenrückseiten betroffen sind und diese nach der ersten Aktion erscheinen.
Ergebnis: Der Fehler wurde behoben.
- **Spielende und Scorefenster:** Das Spiel wird am Ende beendet, ohne dass ein Scorefenster angezeigt wird.
Einstufung: kritisch, da keine Scores eingesehen werden können und auch kein Spiel neu gestartet werden kann. Es kann so kein Gewinner ermittelt werden und es fehlt demnach eine vollständige Funktion des Spiels.
Ergebnis: Der Fehler wurde behoben. Es wird nun ein Scorefenster mit dem Gewinner und den Punkten aller Mitspieler angezeigt. Des weiteren wird eine Umfrage gestartet, ob die Spieler eine weitere Runde spielen möchten. Bestätigen alle Spieler diese positiv, so wird ein neues Spiel gestartet. Ansonsten wird das Spiel aufgelöst und alle Spieler werden in die Lobby zurückgeleitet.
- **GUI Wizard:** Befindet man sich beim Spiel Wizard in der letzten Spielrunde, so gibt es keine Trumpfkarte oder Trumpffarbe mehr. Das Feld für diese beiden Anzeigen sollte also leer sein. Dies ist jedoch nicht der Fall, denn es wird immernoch die Trumpfkarte der vorherigen Runde angezeigt.
Einstufung: wichtig, da das Spiel fehlerfrei gespielt werden kann, jedoch die Karte sehr verwirrend für die Spieler ist und so eventuell falsche Vorhersagen getroffen werden.
Ergebnis: Der Fehler wurde behoben. Das Kartenfeld ist in der letzten Runde leer.

2.3 Während der Validierung gefundene Bugs

Bugs, die während der Validierung durch interaktives Testen des Spiels vom Team oder durch JUnit-Tests gefunden wurden, werden im Folgenden ausgeführt.

- Gefunden durch interaktive Spiel-Tests im Team
 - **Farbzwang-Bug Wizard:** Wenn als erste Karte der Runde ein Narr gelegt wird und anschließend ein Zauberer, muss man die Farbe des Zauberers zugeben, obwohl man jede beliebige Karte spielen können sollte.
Einstufung: kritisch, da das Regelwerk falsche Aktionen fordert. Das Spiel funktioniert dadurch

nicht mehr einwandfrei.

Ergebnis: Der Fehler wurde behoben, indem kleine Änderungen an der `isValidMove`-Methode der Klasse `ClientWizard` und `ServerWizard` vorgenommen wurden.

- Gefunden durch JUnit-Testing

- **Scorefenster-Bug:** Wenn nach einem Spiel das Scorefenster geöffnet wird und der Spieler bei der Umfrage, ob ein neues Spiel gestartet werden soll, auf 'nein' klickt, wird keine Aktion ausgeführt, da im Model kein `ComObject` für diesen Fall versendet wird. Gefunden wurde dieser Bug durch den `ClientViewGameEndTest`.

Einstufung: wichtig, da das Spiel auch auf das Klicken auf 'X' im Spielfeld beendet werden kann, jedoch die Umfrageoption nicht einwandfrei funktioniert.

Ergebnis: Der Fehler wurde behoben. Das entsprechende `ComObject` wird versendet und das Spiel geschlossen.

- **Disconnect-Bug im Server:** Im `GameServer` stand in den Methoden statt `disconnectPlayer(player)` nur `player.closeConnection`. Die Namen der Spieler, die das Spiel verließen, wurden deshalb nicht mehr frei gegeben. Der Fehler wurde durch die `GameServerCheatTests` gefunden.

Einstufung: kritisch, da ein Spieler nach dem Logout seinen Namen wieder benutzen können muss. Des weiteren würde die Liste mit den Spielernamen im Server immer länger werden, was unnötigen Speicherplatz verbraucht.

Ergebnis: Der Fehler wurde behoben, die Namen werden wieder frei gegeben.

3 Plichtenheft Tests

Die im Plichtenheft beschriebenen Tests wurden alle ausgeführt und bestanden. Mit Ausnahme von Test `/T240/`, denn hier wurde eine größere Änderung vorgenommen. Statt über eine Spieleinstellung und eine Pfad-angabe ein neues Hintergrundbild einzufügen, muss der Benutzer das Bild nun in den Ordner 'Data' einfügen, es wird dann automatisch zur Auswahl hinzugefügt.

In folgenden Tests wurde der Text der Fehlermeldung verändert: `/T025/`, `/T027/`, `/T029/`, `/T097/`, `/T280/`, `/T245/`.

In folgenden Tests wurde eine falsche Aktion durch die GUI unterbunden, statt eine Fehlermeldung auszugeben: `/T092/`, `/T117/`, `/T182/`, `/T185/`.

In folgenden Tests wurde eine unnötige Nachricht vom Server im Chat weggelassen: `/T170/`, `/T180/`, `/T228/`, `/T230/`.

Es folgt eine genaue Beschreibung der Testfälle aus dem Plichtenheft mit Testergebnis. Die Angabe der Fehlermeldungen erfolgt in deutsch, jedoch enthalten sie in anderen Sprachen den selben Textinhalt.

- Login (keine Vorraussetzungen)

- `/T020/ Verbindungs Aufbau mit Server:` Mr. Blue gibt den Nickname 'MrBlue' und die Adresse des Servers ein (`/F040/`). Er verbindet sich mit dem Server und gelangt die Lobby (`/F045/`).

Ergebnis: Der Test wurde bestanden.

- `/T025/ Falsche Eingabe (Name):` Mr. Blue gibt keinen Namen ein, es kommt zu einer Fehlermeldung (`/L290/ 'Kein Name angegeben'`).

Ergebnis: Der Test wurde bestanden. Die Fehlermeldung lautet 'Bitte gib einen Benutzernamen ein.'

- `/T027/ Falsche Eingabe (Name):` Nachdem Mr. Blue schon mit dem Name 'MrBlue' auf dem Server eingeloggt ist, versucht nun Mr. White sich mit dem Nickname 'MrBlue' einzuloggen. Es wird eine Fehlermeldung angezeigt (`/L290/ 'Name wird bereits verwendet'`).

Ergebnis: Der Test wurde bestanden. Die Fehlermeldung lautet 'Login fehlgeschlagen. Der Benutzername ist bereits vergeben.'

- `/T029/ Falsche Eingabe (IP):` Mr. Blue gibt eine falsche Serveradresse ein. Beim Loginversuch wird ein Fehler ausgegeben (`/L290/ 'Server konnte nicht gefunden werden'`).

Ergebnis: Der Test wurde bestanden. Die Fehlermeldung lautet 'Diese Adresse ist unbekannt.'

- `/T030/ Auswahl der Sprache:` Mr. Blue wählt im Dropdown-Menü die Sprache 'English' aus (`/F050W/`), das Login-Fenster wird neu erstellt und es und alle folgenden Fenster sind nun in englischer Sprache.

Ergebnis: Der Test wurde bestanden.

- /T035/ *Beenden und wieder Einloggen*: Mr. Blue ist mit dem Namen 'MrBlue' eingeloggt. Er beendet das Spiel und der Nickname wird wieder freigegeben (/L380/). Er startet das Spiel noch einmal und loggt sich wieder mit 'MrBlue' ein (/F045/).
Ergebnis: Der Test wurde bestanden.
- Lobby (Voraussetzungen: /T020/) :
 - /T040/ *Spielerliste*: Der eigene Benutzername 'MrBlue' und die Benutzernamen aller anderen verbundenen Clients werden sofort nach dem Login in der Spielerliste angezeigt (/L100/).
Ergebnis: Der Test wurde bestanden.
 - /T050/ *Serverliste*: Alle von Nutzern erstellten Spiele werden direkt nach deren Erstellung in der Spielliste angezeigt (/L110/).
Ergebnis: Der Test wurde bestanden.
 - /T060/ *Chatnachricht senden*: Mr. Blue gibt die Nachricht 'Hallo Kartenwelt!' in das untere Textfeld ein und drückt die Entertaste (/F060/). Im oberen Textfeld wird die aktuellste Nachricht und alle bisher gesendeten Nachrichten angezeigt (/L115/). Mr. Pink ist auch in der Lobby, sieht die Nachricht von Mr. Blue (/L115/) und sendet die Nachricht 'Hallo MrBlue' (/F060/). Diese erscheint ebenfalls im Chat aller Spieler(/L115/).
Ergebnis: Der Test wurde bestanden.
 - /T080/ *Spiel erstellen*: Mr. White erstellt ein Spiel und gelangt ins Erstellungsfenster (/F080/).
Ergebnis: Der Test wurde bestanden.
 - /T085/ *Mehrere Spiele erstellen (Voraussetzung: /T080/, /T110/)*: Mr Pink erstellt ein weiteres Spiel und gelangt ins Erstellungsfenster (/F080/). Für alle eingeloggten Spieler sind nun beide Spiele sichtbar.
Ergebnis: Der Test wurde bestanden.
 - /T090/ *Spiel beitreten*: Mr. Blue tritt dem bereits korrekt geöffneten und noch nicht vollen Spiel von Mr. White bei und gelangt ins Wartefenster (/F070/).
Ergebnis: Der Test wurde bestanden.
 - /T092/ *Fehler bei Spiel beitreten*: Mr. Brown hat kein Spiel aus der Liste ausgewählt. Er drückt den 'Join'-Knopf und es wird eine Fehlermeldung angezeigt (/L290/ 'Kein Spiel ausgewählt').
Ergebnis: Der Test wurde bestanden. Es wird keine Fehlermeldung angezeigt, solange kein Spiel markiert wurde, ist der Button ausgegraut.
 - /T095/ *Passwortabfrage*: Das Spiel von Mr. White ist passwortgeschützt. Mr. Blue gibt 'youshouldtip' in das zugehörige Textfeld ein (/F140/) und drückt den 'Join'-Knopf, wodurch er zum Wartefenster weitergeleitet wird (/F142/).
Ergebnis: Der Test wurde bestanden.
 - /T097/ *Passwortabfrage(Abbruch)*: Das Spiel von Mr. White ist passwortgeschützt. Mr. Blue kennt das Passwort nicht. Er gibt 'damn' in das Textfeld ein und drückt den 'Join'-Knopf. Eine Fehlermeldung erscheint (/L290/ 'Falsches Passwort') und er bleibt in der Passwortabfrage. Er den 'Cancel'-Knopf und gelangt zurück in die Lobby (/F145/).
Ergebnis: Der Test wurde bestanden. Es wird die Fehlermeldung ausgegeben 'Zutritt verweigert. Das Passwort ist inkorrekt.'
- Erstellungsfenster (Voraussetzung: /T080/):
 - /T100/ *Regelwerk auswählen*: Mr. White wählt 'Wizard' als Regelwerk aus (/F120/) und schaut sich dessen Logo an (/L120/), sowie eine kurze Beschreibung des Spiels, die erscheint, wenn er mit der Maus über das Logo fährt (/L122/).
Ergebnis: Der Test wurde bestanden.
 - /T105/ *Regelwerk auswählen*: Mr. White wählt 'Hearts' als Regelwerk aus (/F120/) und schaut sich dessen Logo an (/L120/), sowie eine kurze Beschreibung des Spiels, die erscheint, wenn er mit der Maus über das Logo fährt (/L122/).
Ergebnis: Der Test wurde bestanden.

- /T110/ *Spiel erstellen*: Mr. White gibt 'NewHeist' als Spielname ein (/F122/). Er setzt einen Haken bei 'Passwort' und gibt 'youshouldtip' als Passwort ein (/F130W/). Er drückt auf den 'Create'-Knopf, erstellt somit ein Spiel und wird zum Wartefenster weitergeleitet (/F126/).
Ergebnis: Der Test wurde bestanden.
- /T115/ *Spiel erstellen*: Mr. White gibt keinen Spielnamen an. Bei drücken des 'Create'-Knopfes wird ein Spiel mit dem Namen 'MrWhite's Spiel' erstellt.
Ergebnis: Der Test wurde bestanden.
- /T117/ *Fehler bei Spiel erstellen*: Mr. White gibt 'NewHeist' als Spielname ein (/F122/). Er setzt einen Haken bei 'Passwort', aber vergisst ein Passwort einzugeben. Beim drücken des 'Create'-Knopfes wird eine Fehlermeldung angezeigt (/L290/ 'Kein Passwort gewählt').
Ergebnis: Der Test wurde bestanden. Solange kein Passwort in das Feld eingegeben wurde, obwohl der Haken gesetzt ist, kann der Erstellen-Button nicht gedrückt werden.
- /T120/ *Erstellung abbrechen*: Mr. White erfährt, dass Mr. Pink schon in einem anderen Spiel ist und entschließt sich die Erstellung abbrechen. Er drückt den 'Cancel'-Knopf und gelangt zurück in die Lobby (/F124/).
Ergebnis: Der Test wurde bestanden.
- Wartefenster (Vorraussetzung: /T110/, /T090/)
 - /T130/ *Chat*: Mr. Pink und Mr. White befinden sich im Wartefenster des Spiels. Mr. Pink schreibt die Chatnachricht 'Hello Mr. White' und sie wird bei beidem im Chat angezeigt (/F160/).
Ergebnis: Der Test wurde bestanden.
 - /T170/ *Verlassen*: Mr. Brown ist gerade dem Spiel beigetreten und verlässt es sofort wieder. Im Chat wird 'MrBrown ist dem Spiel beigetreten' angezeigt (/L270/). Er gelangt zurück in die Lobby (/T170/). Mr. White und Mr. Pink sehen den Namen 'MrBrown' kurz in der Spielerliste, bevor er wieder verschwindet.
Ergebnis: Der Test wurde bestanden. Es wird keine Nachricht im Chat angezeigt, da diese unnötig ist.
 - /T180/ *Spieler entfernen*: Mr. Blue tritt dem Spiel bei. Mr. Pink hat das Spiel erstellt. Er ist somit Spielleiter und entschließt sich dazu Mr. Blue aus dem Spiel zu werfen (/F180/). Nach Drücken des 'Remove Player'-Knopfes gelangt Mr. Blue wieder in die Lobby. Mr. Blue erhält die Nachricht 'Sie wurden vom Spiel entfernt'. Im Chat wird angezeigt 'MrPink hat MrBlue vom Spiel entfernt'(/L270/).
Ergebnis: Der Test wurde bestanden. Im Chat wird keine Nachricht angezeigt, da diese unnötig ist.
 - /T182/ *Fehler bei Spieler entfernen*: Mr. Pink ist Spielleiter. Er drückt den 'Remove Player'-Knopf, hat aber keinen Spieler ausgewählt. Es wird eine Fehlermeldung angezeigt (/L290/ 'Kein Spieler ausgewählt').
Ergebnis: Der Test wurde bestanden. Es wird keine Fehlermeldung angezeigt, sondern der Button ist so lange ausgegraut, wie kein Spieler ausgewählt wurde.
 - /T185/ *Fehler bei Starten des Spiels*: Mr. White ist allein im Wartefenster. Er drückt auf den 'Start'-Knopf und es wird eine Fehlermeldung angezeigt (/L290/ 'Nicht genügend Spieler'). Ergebnis: Der Test wurde bestanden. Es wird keine Fehlermeldung angezeigt, sondern der Button ist so lange ausgegraut, bis genügend Spieler anwesend sind.
 - /T190/ *Spiel starten*: Mr. Brown und Mr. Orange treten dem Spiel bei. Es sind nun genug Spieler vorhanden, deshalb entschließt sich Mr. Pink dazu, das Spiel zu starten (/F200/). Alle Spieler gelangen ins Spielfenster, das Wartefenster wird geschlossen.
Ergebnis: Der Test wurde bestanden.
 - /T200/ *Spiel auflösen*: Es dauert Mr. Pink zu lange, bis genug Spieler zusammenkommen, also verlässt er das Wartefenster. Da er der Spielleiter war, wird das Spiel aufgelöst und alle Spieler gelangen zurück zur Lobby (/F190/).
Ergebnis: Der Test wurde bestanden.

- Spielfenster Allgemein (Voraussetzung: /T190/)

- /T210/ *Spielbeginn* (Voraussetzung: /T100/): Mr. Pink, Mr. White, Mr. Brown und Mr. Orange befinden sich im Spielfenster. Der Server teilt die Karten aus. Jeder sieht seine, ihm vom Server ausgeteilten Karten (/L190/) und die verdeckten Karten seiner Mitspieler (/L192/).
Ergebnis: Der Test wurde bestanden.
- /T220/ *Zug*: (Voraussetzung: /T210/) Mr. Pink ist nicht am Zug, jedoch versucht er dennoch eine Karte zu spielen, was mit einem entsprechendem Dialog unterbunden wird (/L290/).
Ergebnis: Der Test wurde bestanden. Es wird die Fehlermeldung angezeigt 'Deine Aktion stimmt nicht mit der momentanen Spielphase überein.'
- /T240/ *Hintergrund ändern*: Mr. Pink klickt im Dropdown-Menü auf 'Einstellungen'. Ein Dialog mit einer Pfadabfrage öffnet sich. Mr. Pink gibt die Datei './neuerHintergrund.png' im Pfad an. Der Hintergrund des Spielfelds zeigt diese Bild an.
Ergebnis: Der Test wurde bestanden. Der Spieler muss jedoch das Bild in den Ordner 'Data' einfügen. Im Spiel wird es automatisch zur Auswahl angezeigt, wenn der Spieler auf 'Hintergrund' drückt.
- /T270/ *Spielende*: Mr. Orange spielt die letzte Karte des Spiels. Das Spiel ist zu Ende und der Server bestimmt den Gewinner. Mr. Orange bekommt eine Zusammenfassung angezeigt, die ihn als Sieger erklärt und seine Punkte anzeigt(/L250/). Alle anderen Spieler erhalten ebenfalls eine Zusammenstellung ihrer Leistung (/L250/).
Ergebnis: Der Test wurde bestanden. Jeder Spieler bekommt eine Übersicht auf der eingesehen werden kann, welcher Spieler gewonnen hat und wie viele Punkte jeder einzelne Spieler erreicht hat.
- /T280/ *Spielabbruch*: (Voraussetzung /T190/) Es läuft für Mr. Pink nicht sonderlich gut und er entschließt sich, es sein zu lassen, weshalb er das Spiel durch einen Klick auf 'schließen' beendet und sich anschließend in der Lobby wiederfindet (/F210/). Mr. White, Mr. Brown und Mr. Orange werden über das Verlassen des Spielers 'Mr. Pink' durch eine Fehlermeldung informiert (/L290/) und werden ebenfalls in die Lobby zurückgeworfen.
Ergebnis: Der Test wurde bestanden. Die anderen Spieler erhalten die Fehlermeldung 'Das Spiel wurde aufgelöst'.
- /T300/ *Chat*: Mr. Pink, Mr. White, Mr. Brown und Mr. Orange befinden sich im Spielfenster. Mr. Orange schreibt die Chatnachricht 'Hallo zusammen!' (/F220/) und sie wird bei allen im Chat angezeigt (/L260/).
Ergebnis: Der Test wurde bestanden.
- /T310/ *Chat*: Mr. Pink, Mr. White, Mr. Brown und Mr. Orange befinden sich im Spielfenster. Mr. Orange versucht eine leere Nachricht im Chatfenster abzusetzen, die Nachricht wird nicht gesendet und erscheint auch nicht in seinem Chat.
Ergebnis: Der Test wurde bestanden.
- /T320/ *Neue Runde*: Das Spiel wurde fertig gespielt. Jeder Mitspieler erhält eine Abfrage ('Nochmal spielen?') (/L275W/). Jeder Spieler stimmt mit 'Ja'. Es wird ein neues Spiel gestartet (/F200/).
Ergebnis: Der Test wurde bestanden. Die Abfrage befindet sich im selben Fenster, wie Übersicht am Ende eines Spiels.

- Spielfenster Wizard

- /T228/ *Rundenbeginn*: (Voraussetzung: /T210/) Der Server dreht die oberste Karte auf dem Aufnahmestapel (/L195/) automatisch um und sagt dadurch die Trumpffarbe (ausgenommen einen Zauberer) an. Im Chat erscheint eine Spielnachricht des Servers (/L270/).
Ergebnis: Der Test wurde bestanden. Es wird keine Nachricht im Chat angezeigt, da diese unnötig ist.
- /T230/ *Stiche ansagen*: (Voraussetzung: /T228/) Nach längerem Spiel wird einer der 4 Zauberer vom Server beim Kartengeben aufgedeckt und da Mr. Pink an der Reihe ist, erscheint ein Auswahlfenster. Mr. Pink wählt die Trumpffarbe. (/F360/) Über den Spielchat erhalten alle Spieler Auskunft über den Vorgang (/L270/) und sehen die angesagte Farbe.
Ergebnis: Der Test wurde bestanden. Es wird keine Chatnachricht gesendet, sondern die Farbe wird in dem zusätzlichen Feld für die Trumpffarbe angezeigt.

- /T240/ *Stiche ansagen: (Voraussetzung: /T228/ oder /T230/)* Nachdem eine Trumpffarbe angesagt wurde erscheint ein Fenster, indem jeder Spieler die Anzahl seiner Stiche tippen kann. Mr. Brown glaubt 4 Stiche machen zu können und gibt die Zahl ein (/F370/). Auch die anderen Spieler geben ihre Tipps ab und sehen die prognostizierten Stiche der anderen.(/L196/).
Ergebnis: Der Test wurde bestanden.
- /T245/ *Stiche ansagen: (Voraussetzung: /T228/ oder /T230/)* Nachdem eine Trumpffarbe angesagt wurde erscheint ein Fenster, indem jeder Spieler die Anzahl seiner Stiche tippen kann. Mr. Orange gibt 'zehn' in das Feld ein. Es wird eine Fehlermeldung angezeigt (/L290/ 'Bitte geben sie eine Zahl ein').
Ergebnis: Der Test wurde bestanden. Es wird die Fehlermeldung 'Diese Zahl ist nicht erlaubt.' angezeigt.
- /T260/ *Rundenende:* Mr. Pink spielt die letzte Karte der Runde. Es erscheint eine kurze Zusammenfassung, wie viele Stiche er gemacht hat, wie viele er getippt hat und erhält seinen aktuellen Punktestand sowie den der anderen Mitspieler in einem separaten Fenster angezeigt (/L250/).
Ergebnis: Der Test wurde bestanden. Jeder Spieler hat ein Feld, in dem die oben genannten Informationen angezeigt werden.
- Spielfenster Hearts
 - /T290/ *Spielbeginn Hearts: (Voraussetzung /T105/)* Mr. Pink, Mr. White, Mr. Orange und Mr. Brown befinden sich im Spielfenster. Das Spiel beginnt und ein Fenster erscheint bei allen Spielern in dem 3 Karten ausgewählt werden müssen. Mr. White wählt 3 Karten aus und bestätigt seine Auswahl (/F260/). Nachdem Mr. Orange ebenfalls 3 Karten zur Weitergabe ausgewählt hat erhält diese Mr. White. Mr. Pink erhält die 3 Karten von Mr. White und Mr. Orange erhält die Karten von Mr. Pink.
Ergebnis: Der Test wurde bestanden.
 - /T295/ *Spielbeginn Hearts: (Voraussetzung /T105/)* Mr. Pink, Mr. White, Mr. Orange und Mr. Brown befinden sich im Spielfenster. Das Spiel beginnt und ein Fenster erscheint bei allen Spielern, in dem 3 Karten ausgewählt werden müssen. Mr. White nur 2 Karten aus und bestätigt seine Auswahl (/F260/). Es erscheint ein Dialog der ihn auffordert 3 Karten auszuwählen(/L290/).
Ergebnis: Der Test wurde bestanden. Die Fehlermeldung lautet 'Du kannst diese Karten nicht tauschen.'.
 - /T330/ *Zug: (Voraussetzung /T210/)* Mr. Pink versucht eine ungültige Karte zuzugeben. Das Spiel erkennt den Regelverstoß und zeigt dies über einen Dialog an (/L290/).
Ergebnis: Der Test wurde bestanden. Die Fehlermeldung lautet 'Der Zug ist nicht gültig!'.

4 Line- und Branch-Coverage

4.1 View

Die View wurde in erster Linie durch einen interaktiven Test getestet. Auch über JUnit-Tests gibt es für die View Ergebnisse, die jedoch nicht ausreichend sind, da nicht versucht wurde eine möglichst hohe Abdeckung zu erreichen. Der Interaktive Test ist somit der aussagekräftigere Test und wird im Anschluss ebenfalls zu den Ergebnissen der JUnit-Tests aus den anderen Paketen gerechnet.

4.1.1 Interaktiver Test

- Betrachtete Testfälle: /T020/, /T025/, /T027/, /T029/, /T030/, /T040/, /T050/, /T060/, /T080/, /T085/, /T090/, /T095/, /T097/, /T100/, /T105/, /T110/, /T115/, /T120/, /T130/, /T170/, /T180/, /T190/, /T200/, /T210/, /T220/, /T240/, /T280/, /T300/, /T310/, /T228/, W/T240/, /T245/, /T250/, /T290/, /T295/, /T330/
- Nicht betrachtete Testfälle:

- Fehlermeldung durch ausgegrauten Button ersetzt: /T092/, /T117/, /T182/, /T185/.
- Testfall, der das Programm beenden und somit die Coverage-Aufzeichnung unterbrechen würde: /T035/.
- Vom Zufall abhängig (eigentlich deswegen auch kein Testfall): /T230/.
- Setzen Ende des Spiels voraus, Zeitaufwand zu hoch: /T270/, /T320/, /T260/.

Paket	Klassen	Line Coverage in %	Branch Coverage in %
Client.View	20	90,6	72,5

Klassen in diesem Paket	Line Coverage in %	Branch Coverage in %
GamePanel	96,50	93,50
ScoreWindow	64,10	31,20
ChooseItem	65,10	33,30
Game	96,60	78,60
ViewCard	89,60	83,30
CreateGame	86,20	58,60
Lobby	93,90	73,30
ChooseCards	91,90	66,70
Warning	82,20	400
DiscardPile	70,60	-
TrumpColour	78,30	57,10
GameLobby	98,50	87,50
InputNumber	93,50	75,00
Language	100,00	-
DrawDeck	83,30	-
OtherPlayer	96,90	100,00
Login	92,90	66,70
Password	97,00	71,40
OwnHand	100,00	100,00
OwnOtherData	100,00	100,00

Resultate vom interaktiven Test bei restlichen Paketen:

Klasse	Line Coverage in %	Branch Coverage in %
Ruleset	5,40	3,70
Server	-	-
Client	34,10	37,90
ComObjects	36,80	-
Gesamt	22,20	17,20

4.1.2 JUnitTest-Ergebnisse

Die Ergebnisse ergeben sich aus wenigen Tests, die eine View öffnen. Sie können nicht als aussagekräftige Ergebnisse gezählt werden, da in den JUnit-Tests für dieses Paket keine möglichst große Abdeckung angestrebt wurde. Diese soll durch einen interaktiven Test erreicht werden (siehe 4.1.1).

Paket	Klassen	Line Coverage in %	Branch Coverage in %
Client.View	63	59,00	42,00

Klassen in diesem Paket	Line Coverage in %	Branch Coverage in %
ChooseCards	78,00	66,00
ChooseCards\$1	0,00	-
ChooseCards\$2	0,00	-
ChooseCards\$3	0,00	-
ChooseCards\$4	100,00	-
ChooseCards\$CardMouseListener	0,00	0,00
ChooseItem	69,00	50,00
ChooseItem\$1	0,00	-
CreateGame	60,00	44,00
CreateGame\$1	0,00	0,00
CreateGame\$2	0,00	-
CreateGame\$3	100,00	-
CreateGame\$LeaveButtonListener	0,00	-
CreateGame\$MouseOverListener	0,00	-
CreateGame\$PWCheckedListener	0,00	0,00
CreateGame\$PWEmptyListener	0,00	0,00
CreateGame\$RulesetSelectionListener	100,00	-
DiscardPile	70,00	-
DrawDeck	83,00	-
Game	70,00	51,00
Game\$1	64,00	60,00
Game\$2	100,00	-
Game\$3	0,00	-
Game\$4	0,00	-
Game\$5	40,00	50,00
Game\$6	0,00	-
Game\$bgMenuAction	0,00	-
Game\$cardsMenuAction	0,00	-
GameLobby	71,00	44,00
GameLobby\$1	91,00	66,00
GameLobby\$2	71,00	50,00
GameLobby\$3	0,00	-
GameLobby\$PlayerSelectedListener	0,00	0,00
GamePanel	55,00	41,00
InputNumber	67,00	66,00
InputNumber\$1	0,00	-
InputNumber\$2	0,00	-
Language	100,00	-
Lobby	59,00	28,00
Lobby\$1	0,00	-
Lobby\$2	100,00	-
Lobby\$3	0,00	-
Lobby\$4	100,00	-
Lobby\$5	0,00	-
Lobby\$GameSelectedListener	0,00	0,00
Login	73,00	63,00
Login\$1	0,00	-
Login\$NameLengthListener	0,00	0,00
Login\$NameLengthListener\$1	0,00	-
OtherPlayer	62,00	50,00
OwnHand	92,00	100,00
OwnOtherData	66,00	50,00

Klassen in diesem Paket	Line Coverage in %	Branch Coverage in %
Password	58,00	36,00
Password\$1	0,00	-
Password\$LeaveButtonListener	0,00	-
ScoreWindow	66,00	41,00
ScoreWindow\$1	80,00	50,00
ScoreWindow\$2	100,00	-
TrumpColour	16,00	0,00
ViewCard	43,00	16,00
Warning	79,00	60,00
Warning\$1	0,00	-
Warning\$CloseListener	0,00	0,00

4.2 Client

Die Werte wurden durch JUnit-Testing erreicht.

Paket	Klassen	Line Coverage in %	Branch Coverage in %
Client	32	56,00	56,00

Der Grund, warum die Gesamt-Coverage-Werte nur bei etwas über 50% liegen, ist, dass in diesem Paket sehr viele Standard API-Klassen zu finden sind, die nicht explizit getestet wurden. Vernachlässigt man sämtliche Client.Controller-Klassen und die Main-Klassen, sowie die Thread-Klasse `MessengerThread`, werden sehr hohe Werte erzielt. Der Wert des `LanguageInterpreters` ist deshalb so niedrig, weil es sich dabei um eine Klasse handelt, die fast ausschließlich swich-case-Abfragen enthält, die nicht explizit getestet wurden.

Klassen in diesem Paket	Line Coverage in %	Branch Coverage in %
ClientController	79,00	50,00
ClientController\$CardMouseListener	0,00	0,00
ClientController\$ChatListenerGame	0,00	0,00
ClientController\$ChatListenerGameLobby	0,00	0,00
ClientController\$ChatListenerLobby	0,00	0,00
ClientController\$ConnectButtonListener	0,00	-
ClientController\$CreateButtonListener	0,00	0,00
ClientController\$GameCloseListener	0,00	-
ClientController\$GameLobbyCloseListener	12,00	-
ClientController\$HostButtonListener	0,00	-
ClientController\$JoinButtonListenerLobby	0,00	0,00
ClientController\$JoinButtonListenerPassword	0,00	-
ClientController\$KickPlayerButtonListener	0,00	-
ClientController\$LanguageSelectionListener	0,00	-
ClientController\$LeaveButtonListenerGameLobby	0,00	-
ClientController\$LeaveButtonListenerLobby	0,00	-
ClientController\$LeaveButtonListenerPassword	0,00	-
ClientController\$LeaveButtonListenerScoreWindow	0,00	-
ClientController\$LobbyCloseListener	12,00	-
ClientController\$OKButtonListenerChooseCards	0,00	-
ClientController\$OKButtonListenerChooseColour	0,00	-
ClientController\$OKButtonListenerInputNumber	0,00	0,00
ClientController\$OKButtonListenerWarning	0,00	0,00
ClientController\$PlayAgainButtonListenerScoreWindow	0,00	-
ClientController\$StartGameButtonListener	0,00	-
ClientMain	0,00	-
ClientMain\$1	0,00	-
ClientModel	89,00	81,00
ClientState	93,00	-
LanguageInterpreter	25,00	26,00
MessageListenerThread	0,00	0,00
ViewNotification	100,00	-

4.3 Server

Die Werte wurden durch JUnit-Testing erreicht.

Paket	Klassen	Line Coverage in %	Branch Coverage in %
Server	8	76,00	75,00

Der Grund, warum die Gesamt-Coverage-Werte nur bei etwa 75% liegen, ist, dass in diesem Paket die Player-Klasse, vorhanden ist, welche durch die JUnit-Tests kaum abgedeckt wurde, da diese Klasse für die Test durch eine Mockklasse TestPlayer ersetzt wurde. Des weiteren existiert eine Main-Klasse, die ebenfalls nicht zu testen war.

Klassen in diesem Paket	Line Coverage in %	Branch Coverage in %
GameServer	86,00	79,00
GameServerRepresentation	-	-
LobbyServer	90,00	84,00
LoginServer	91,00	75,00
LoginServer\$ClientListenerThread	50,00	50,00
Player	13,00	0,00
Server	83,00	80,00
ServerMain	0,00	0,00

4.4 Ruleset

Die Werte wurden durch JUnit-Testing erreicht.

Paket	Klassen	Line Coverage in %	Branch Coverage in %
Ruleset	24	89,00	84,00

Der Grund, warum die Gesamt-Coverage-Werte nur bei etwas über 85% liegen, ist, dass in diesem Paket mehrere Enum- und Exception-Klassen vorhanden sind, die nicht explizit getestet wurden. Des weiteren gibt es viele Klassen, z.B. GameClientUpdate, die nahezu ausschließlich getter und setter Methoden enthalten, für die sich kein Branch-Coverage Wert ergibt.

Klassen in diesem Paket	Line Coverage in %	Branch Coverage in %
Card	-	-
CardComparator	99,00	87,00
ClientHearts	100,00	100,00
ClientRuleset	65,00	54,00
ClientWizard	100,00	90,00
Colour	66,00	-
DiscardedCard	66,00	-
EmptyCard	87,00	-
GameClientUpdate	100,00	-
GamePhase	95,00	-
GameState	82,00	76,00
HeartsCard	100,00	-
HeartsData	78,00	87,00
IllegalNumberOfPlayersException	-	-
OtherData	80,00	100,00
PlayerState	84,00	75,00
RulesetException	-	-
RulesetType	100,00	-
ServerHearts	92,00	86,00
ServerRuleset	87,00	92,00
ServerWizard	90,00	83,00
UserMessages	100,00	-
WizData	100,00	-
WizardCard	100,00	-

4.5 ComObjects

Die Werte wurden durch JUnit-Testing erreicht.

Paket	Klassen	Line Coverage in %	Branch Coverage in %
ComObjects	31	62	-

Der Grund, warum die Gesamt-Coverage-Werte nur bei etwas über 60% liegen und es keine Branch-Coverage-Werte gibt, ist, dass die ComObjects nur zum Weiterreichen von Information gedacht sind und demnach nahezu ausschließlich aus getter- und setter-Methoden bestehen, für die es keine Branch-Coverage Werte gibt. Je nach dem, welche Objects in einem Test benötigt wurden, ist die Line-Coverage mal höher, mal niedriger.

Klassen in diesem Paket	Line Coverage in %	Branch Coverage in %
ComBeenKicked	0,00	-
ComChatMessage	100,00	-
ComClientLeave	66,00	-
ComClientQuit	50,00	-
ComCreateGameRequest	66,00	-
ComInitGameLobby	50,00	-
ComInitLobby	50,00	-
ComJoinRequest	66,00	-
ComKickPlayerRequest	66,00	-
ComLobbyUpdateGamelist	50,00	-
ComLoginRequest	66,00	-
ComNewRound	66,00	-
ComObject	-	-
ComRuleset	100,00	-
ComServerAcknowledgement	0,00	-
ComStartGame	100,00	-
ComUpdatePlayerlist	50,00	-
ComWarning	50,00	-
MsgBoolean	50,00	-
MsgCard	75,00	-
MsgCardRequest	50,00	-
MsgGameEnd	50,00	-
MsgMultiCards	100,00	-
MsgMultiCardsRequest	50,00	-
MsgNumber	100,00	-
MsgNumberRequest	33,00	-
MsgSelection	100,00	-
MsgSelectionRequest	50,00	-
MsgUser	50,00	-
RulesetMessage	-	-

4.6 Alle Pakete und Klassen

Die Werte wurden durch JUnit-Testing erreicht.

Paket	Klassen	Line Coverage in %	Branch Coverage in %
Client	32	56,00	56,00
Client.View	63	59,00	42,00
Client.View interaktiv	20	90,6	72,5
ComObjects	31	62,00	-
Ruleset	24	89,00	84,00
Server	8	76,00	75,00
Alle Pakete*	158	70,00	68,00
Alle Pakete**	158	80,06	73,80

* mit dem Paket Client.View

* mit dem Paket Client.View interaktiv

5 Korrektheitstests

Zur Suche von Fehlern, wurde das Eclipse-Plugin 'FindBugs' verwendet. Es konnte damit ein Fehler gefunden werden. Fehler: Eine If-Abfrage im Regelwerk wurde fälschlicherweise mit einem ';' beendet, sodass der Body leer war und der Code nicht durch die Abfrage ausgeführt wurde, sondern immer. Da dies ein Sonderfall im Regelwerk war, wurde der Fehler nicht anders entdeckt.

6 Skalierungs- und Belastungs-Tests

In diesem Test ging es darum, das Programm auf seine Belastbarkeit zu testen, dazu wurden JUnit-Tests mit Mockklassen erstellt. Des weiteren wurden interessante Werte bzgl. Speicherverbrauch, CPU-Auslastung und Rechenzeit ausgelesen.

Die Tests wurden auf einem Rechner mit folgenden Daten ausgeführt: Prozessor Intel(R) Core(TM) i5-2410M 2.30 GHz mit 64-Bit Betriebssystem, Windows 7.

6.1 Lobby

In der Lobby wurde eine Belastung von zunächst 500 Spielern, dann 1000 Spielern und zuletzt 2000 Spielern getestet. Das Programm hielt allen Tests stand.

CPU-Auslastung (beim Test maximal erreichte Werte):

500 Spieler	ca. 43%
1000 Spieler	ca. 45%
2000 Spieler	ca. 49%

6.1.1 Werte

Erläuterungen zu den Tabellen:

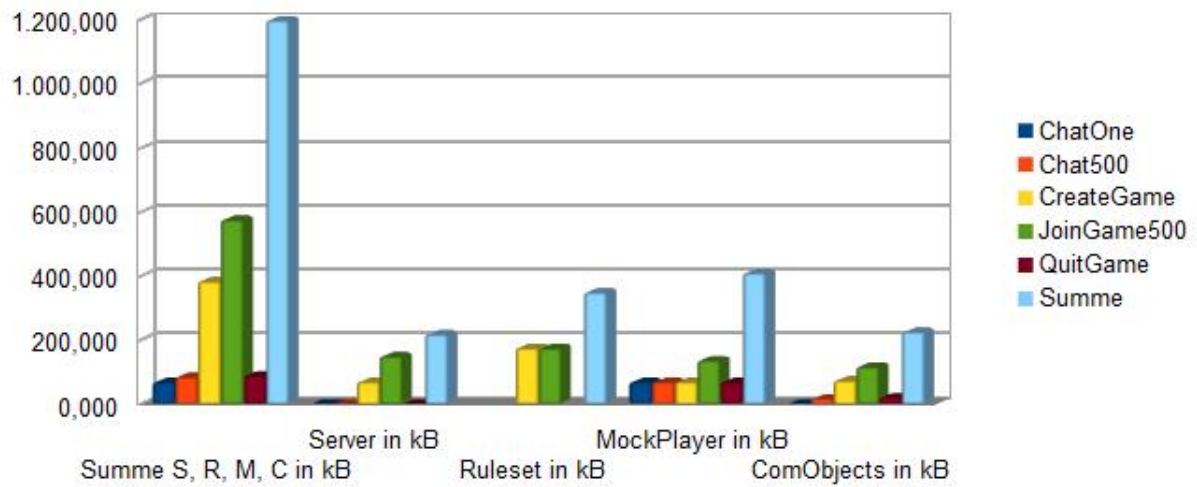
LobbyXSpieler	X Spieler sind in der Lobby
ChatOne	Eine Person sendet eine Nachricht an alle anderen
ChatX	X Personen senden eine Nachricht an alle anderen
Create Game	Alle Spieler in der Lobby erstellen ein Spiel
Join GameX	X Spieler erstellen ein Spiel und X Spieler treten einem bei
QuitGame	Alle Spieler in der Lobby verlassen dass Spiel

Auffälligkeiten bei den Werten:

- Bei ChatOne, ChatX und QuitGame gibt es kein Ruleset, da kein Spiel erstellt wird, deswegen sind die Werte 0.
- Der Speicherplatz für MockPlayer ist immer gleich, außer bei JoinGame, da es noch einmal soviele Game-master für die Spiele gibt.
- Der Server Speicher ist bei CreateGame und JoinGame höher, da viele GameServerRepräsentations an die Spieler verschickt werden.

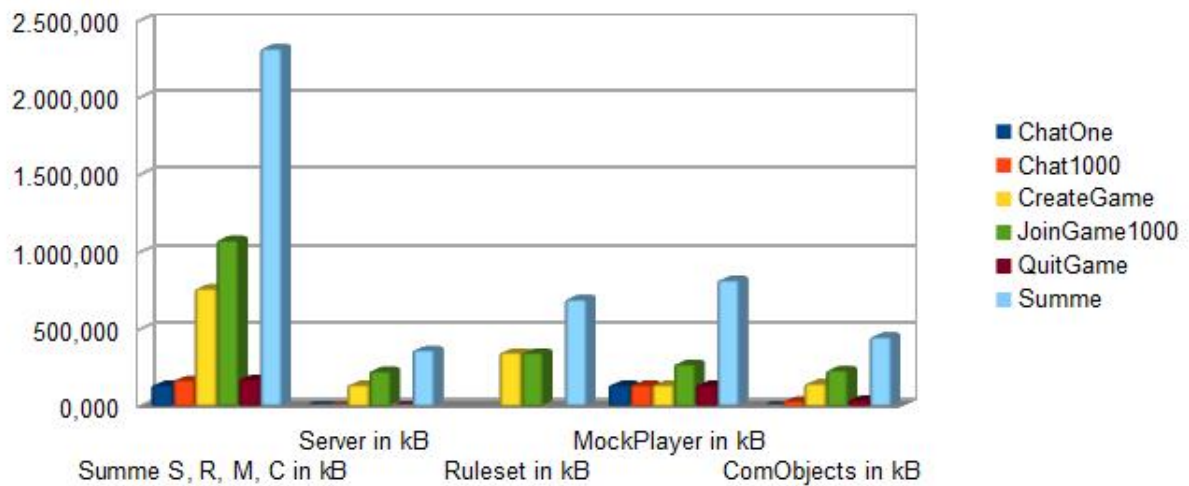
Lobby500Spieler	ChatOne	Chat500	CreateGame	JoinGame500	QuitGame	Summe
Durchschnittszeit in s	0,028	0,101	0,098	0,185	0,051	0,463
Gesamtspeicher in kB	17.846,333	22.053,000	25.199,000	37.967,000	19.599,667	122.665,000
Summe S, R, M, C in kB	68,080	84,048	382,048	574,000	88,048	1.196,224
Server in kB	0,048	0,048	68,048	148,000	0,048	216,192
Ruleset in kB	0,000	0,000	174,000	174,000	0,000	348,000
MockPlayer in kB	68,000	68,000	68,000	136,000	68,000	408,000
ComObjects in kB	0,032	16,000	72,000	116,000	20,000	224,032

Lobby 500 Spieler



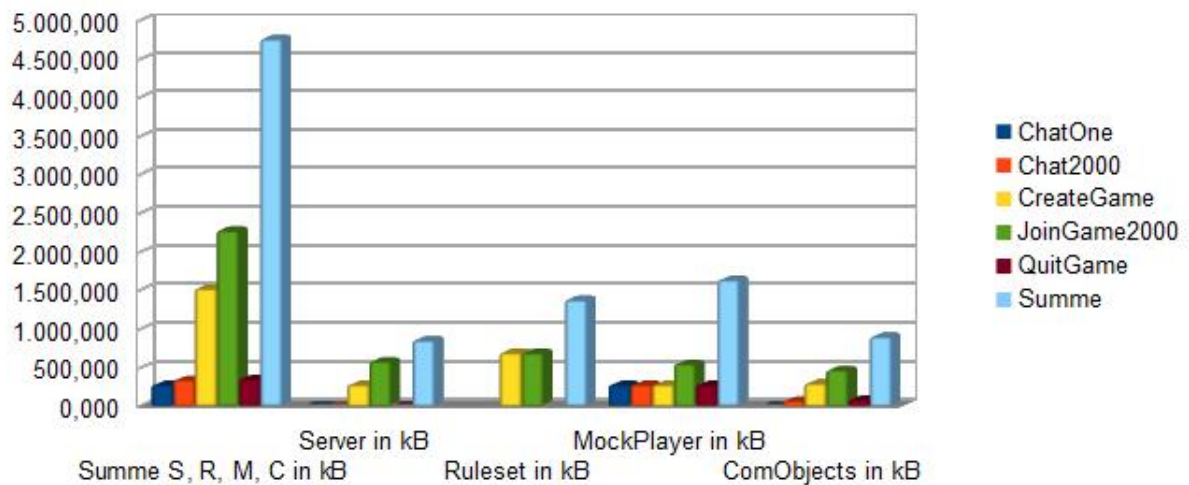
Lobby1000Spieler	ChatOne	Chat1000	CreateGame	JoinGame1000	QuitGame	Summe
Durchschnittszeit in s	0,052	0,226	0,191	0,601	0,104	1,173
Gesamtsspeicher in kB	18.279,667	34.865,000	40.458,667	59.965,333	26.502,000	180.070,667
Summe S, R, M, C in kB	136,080	168,048	762,000	1.076,000	176,048	2.318,176
Server in kB	0,048	0,048	136,000	226,000	0,048	362,144
Ruleset in kB	0,000	0,000	346,000	346,000	0,000	692,000
MockPlayer in kB	136,000	136,000	136,000	272,000	136,000	816,000
ComObjects in kB	0,032	32,000	144,000	232,000	40,000	448,032

Lobby 1000 Spieler



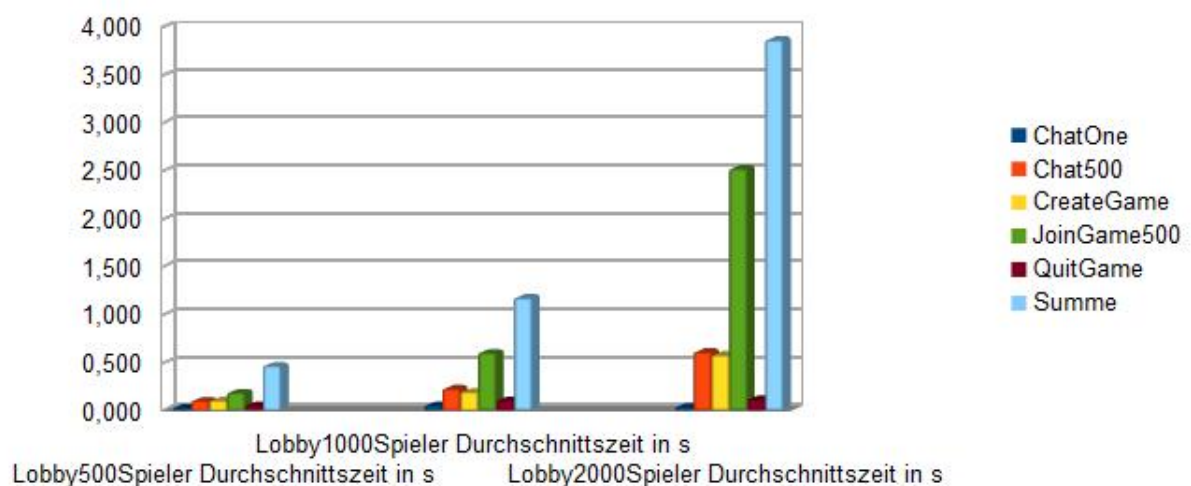
Lobby2000Spieler	ChatOne	Chat2000	CreateGame	JoinGame2000	QuitGame	Summe
Durchschnittszeit in s	0,038	0,610	0,577	2,520	0,117	3,861
Gesamtpeicher in kB	18.932,000	46.328,667	77.498,333	131.333,333	32.925,333	307.017,667
Summe S, R, M, C in kB	272,080	336,048	1.520,000	2.273,000	352,048	4.753,176
Server in kB	0,048	0,048	272,000	577,000	0,048	849,144
Ruleset in kB	0,000	0,000	688,000	688,000	0,000	1.376,000
MockPlayer in kB	272,000	272,000	272,000	544,000	272,000	1.632,000
ComObjects in kB	0,032	64,000	288,000	464,000	80,000	896,032

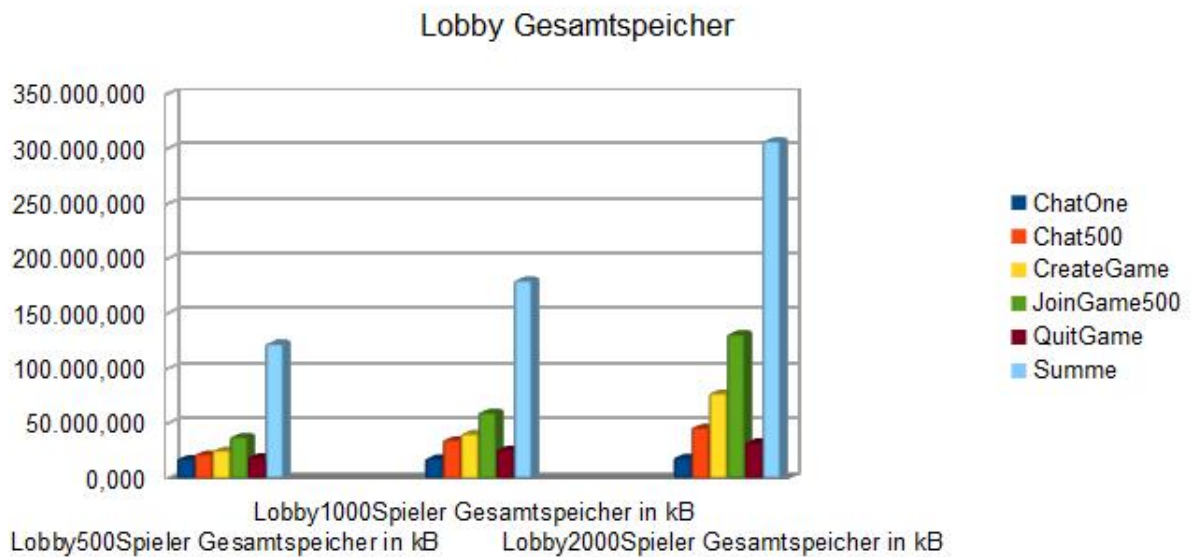
Lobby 2000 Spieler



6.1.2 Zusammenfassung

Lobby Durchschnittszeit





JoinGame braucht die meiste Zeit und Gesamtspeicher, da vorher noch die entsprechende Zahl an Spielen erstellt werden muss.

6.2 Server

Im Server wurde eine Belastung von zunächst 100 Spielen, dann 250 Spielen und zuletzt 500 Spielen mit je 3 Spielern pro Spiel getestet. Das Programm hielt allen Tests stand.

CPU-Auslastung (beim Test maximal erreichte Werte):

100 Spiele	ca. 40%
250 Spiele	ca. 45%
500 Spieler	ca. 54%

6.2.1 Werte

Erläuterungen zu den Tabellen:

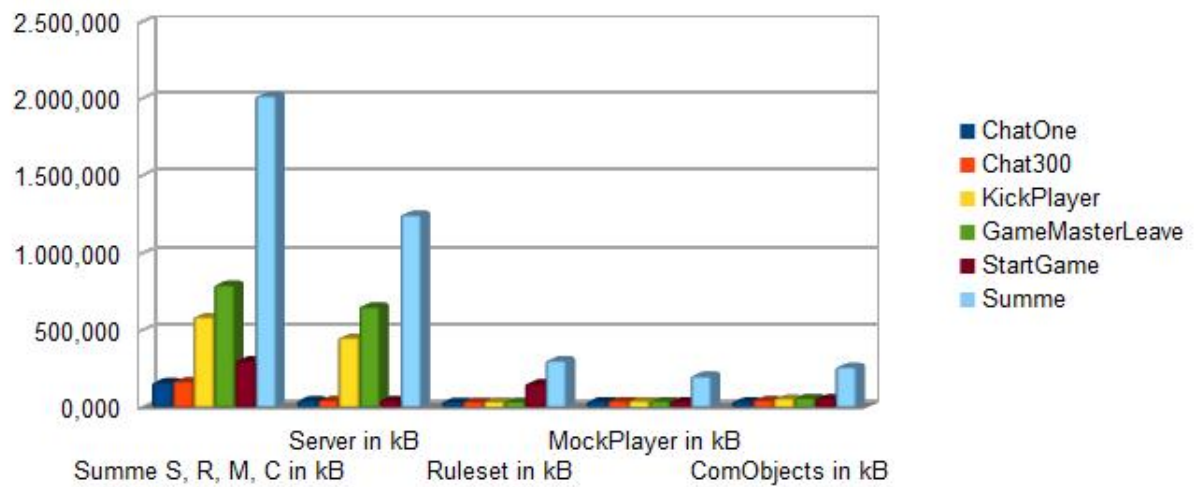
GameServerX	X Spiele mit je 3 Spielern darin
ChatOne	Eine Person in jedem Spiel sendet eine Nachricht an seine Mitspieler
ChatX	Alle Spieler in einem Spiel senden eine Nachricht an alle anderen Spieler im Spiel
KickPlayer	Der GameMaster jedes Spieles wirft einen Spieler raus
GameMasterLeav	Der GameMaster in jedem Spiel verlässt dieses
StartGame	Alle Spiele werden gestartet

Auffälligkeiten bei den Werten:

- Der Server Speicher ist bei KickPlayer und GameMasterLeave sehr viel höher, da viele GameServerRepräsentations an die Spieler verschickt werden.
- Das Ruleset ist bei StartGame am höchsten, da es das Spiel startet und Karten etc. verschickt.
- Bei vielen Spielern verbraucht der Server exponentiell viel mehr Speicher, da sehr viele GameServerUpdates an alle Spieler verschickt werden und diese viel Speicher benötigen.

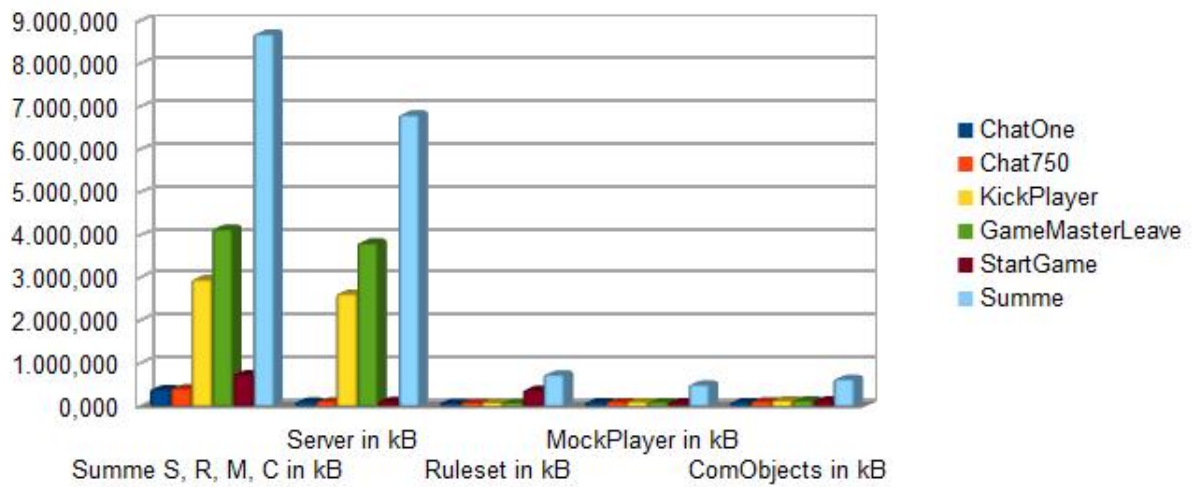
GameServer100	ChatOne	Chat300	KickPlayer	GameMasterLeave	StartGame	Summe
Durchschnittszeit in s	0,126	0,158	0,169	0,191	0,169	0,812
Gesamtspeicher in kB	20.341,000	21.002,333	21.290,000	23.180,667	20.965,000	106.779,000
Summe S, R, M, C in kB	163,792	170,192	586,256	794,656	304,576	2.019,472
Server in kB	45,656	45,656	453,000	655,000	49,656	1.248,968
Ruleset in kB	37,336	37,336	37,336	37,336	155,000	304,344
MockPlayer in kB	40,800	40,800	40,800	40,800	40,800	204,000
ComObjects in kB	40,000	46,400	55,120	61,520	59,120	262,160

Server 100 Spiele



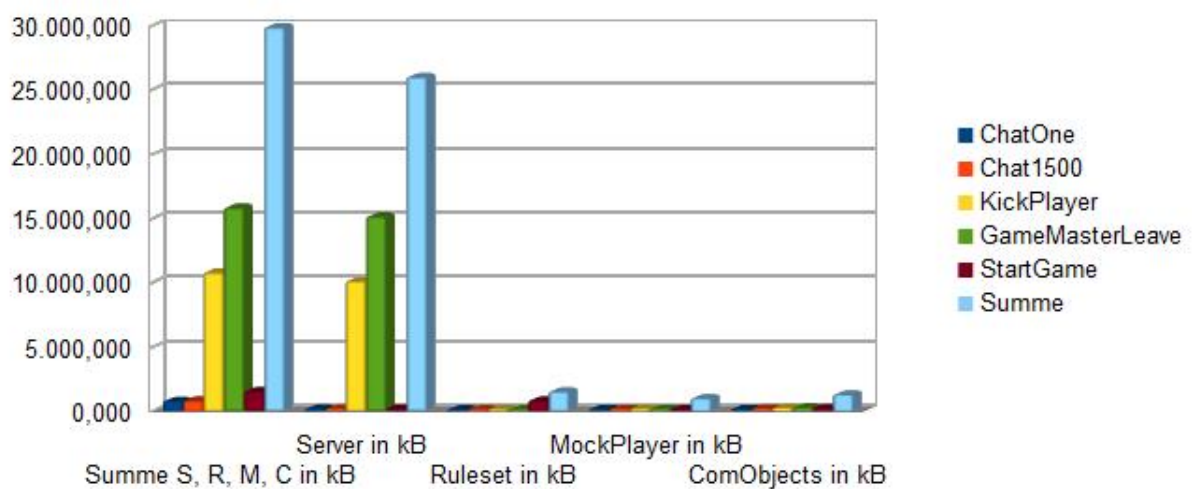
GameServer250	ChatOne	Chat750	KickPlayer	GameMasterLeave	StartGame	Summe
Durchschnittszeit in s	0,225	0,291	0,315	0,419	0,252	1,503
Gesamtspeicher in kB	31.476,333	34.668,000	42.475,333	53.536,667	31.783,667	193.940,000
Summe S, R, M, C in kB	404,936	420,936	2.962,936	4.148,936	755,000	8.692,744
Server in kB	114,000	114,000	2.634,000	3.820,000	124,000	6.806,000
Ruleset in kB	88,936	88,936	88,936	88,936	384,000	739,744
MockPlayer in kB	102,000	102,000	102,000	102,000	102,000	510,000
ComObjects in kB	100,000	116,000	138,000	138,000	145,000	637,000

Server 250 Spiele

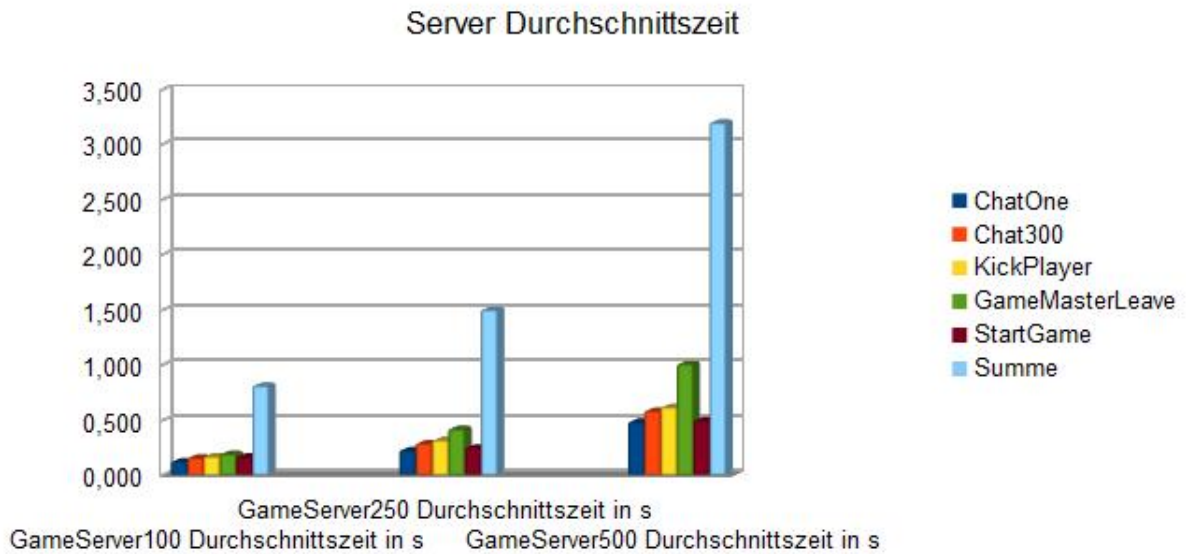


GameServer500	ChatOne	Chat1500	KickPlayer	GameMasterLeave	StartGame	Summe
Durchschnittszeit in s	0,485	0,583	0,620	1,011	0,501	3,199
Gesamtsspeicher in kB	42.339,000	45.215,667	74.373,333	91.421,667	45.612,333	298.962,000
Summe S, R, M, C in kB	806,000	838,000	10.801,000	15.844,000	1.590,000	29.879,000
Server in kB	228,000	228,000	10.147,000	15.158,000	248,000	26.009,000
Ruleset in kB	174,000	174,000	174,000	174,000	846,000	1.542,000
MockPlayer in kB	204,000	204,000	204,000	204,000	204,000	1.020,000
ComObjects in kB	200,000	232,000	276,000	308,000	292,000	1.308,000

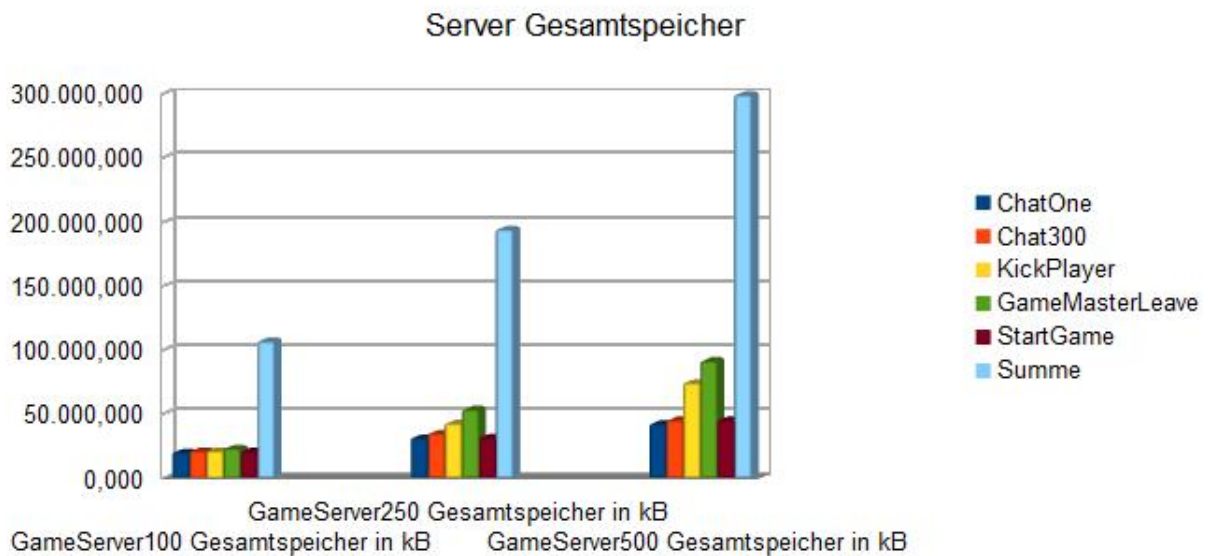
Server 500 Spiele



6.2.2 Zusammenfassung



GameMasterLeave braucht die meiste Zeit und Gesamtspeicher, da alle Spieler die GameLobby verlassen und entsprechende Updates verschickt werden müssen.



7 Weitere Tests

7.1 BETA-Test

Es wurde ein BETA-Test über ein Wochenende gestartet. Es nahmen 15 unabhängige Personen daran teil. Jede Anmerkung wurde ausgewertet und nach Möglichkeit verbessert oder umgesetzt.

- **Doppelter Name durch Ausnutzen von Leerzeichen:** Es bestand die Möglichkeit sich mit dem Namen 'TestSpieler' einzuloggen und zusätzlich noch einmal mit dem Namen 'Testspieler ', also mit einem Leerzeichen zu Beginn oder am Ende des Namens. Die beiden Spieler können so im Spiel nicht direkt

voneinander unterschieden werden, da sie auf den ersten Blick, den selben Nicknamen haben. Die doppelte Nutzung eines Namens durch Anfügen von Leerzeichen wurde unterbunden.

- **Handbuch:** Die Erklärung zum Serverstart ist nicht ausreichend genug. Das Kapitel Installation wurde deshalb überarbeitet.
- **Rundenanzahl:** Es wurde angemerkt, dass die Anzeige einer Rundenzahl sinnvoll wäre, sodass man sehen kann in welcher man sich befindet, ohne mitzuzählen. Die Rundenzahl wird nun zusätzlich im Infofenster angezeigt, in dem der Spieler auch seine anderen Informationen sehen kann (siehe 2.1).
- **Beschränkte Länge des Namens:** Da die Länge des Spielernamens zuvor nicht eingeschränkt war, und deshalb nicht der ganze Name im Spielfenster dargestellt werden konnte, wurde eine Begrenzung der Länge eingeführt.

8 Fakten zum Programm NETWizHearts

- 229 Klassen insgesamt
 - 114 Kernklassen
 - 68 anonyme Klassen
 - 42 Testklassen
 - 5 MockObjekte
- 18611 Zeilen gesamt
 - 14936 Zeilen Code
 - * 7811 Zeilen Production Code
 - * 7125 Zeilen Testcode in 295 Testfällen
 - 3675 Zeilen JavaDoc
- Code-Testcode Verhältnis ca. 1,1:1