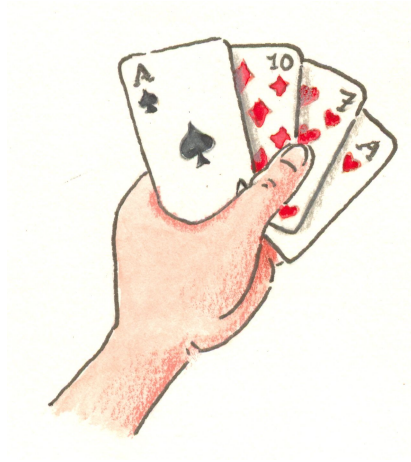


Implementierungsbericht

17. Dezember 2013



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
Verifikation	Patrick Kubin	kubin@fim.uni-passau.de
Präsentation	w	w

Inhaltsverzeichnis

1	Einleitung	2
2	Änderungen gegenüber der Spezifikation	2
2.1	Client	2
2.2	View	2
2.3	Ruleset	2
2.4	Server	3
2.5	ComObjects	7
3	Vergleich: Implementierungsplan und Realität	8
3.1	Änderungen am Plan	8
3.2	Milestone 1	8
3.3	Milestone 2	8
3.4	Milestone 3	9
3.5	Finale Phase	9
4	Vergleich: Implementierung und Pflichtenheft	10
5	Bekannte Bugs	11

1 Einleitung

Dieses Dokument ist eine Dokumentation der Implementierungsphase. In Folgenden werden Änderungen aufgezählt, die seit der letzten Phase entstanden sind, des Weiteren wird eine Übersicht gegeben über die zuvor festgelegten Meilensteine und die tatsächliche Umsetzung dieser Aufteilung.

2 Änderungen gegenüber der Spezifikation

2.1 Client

- **LanguageInterpreter:** *Neu hinzugefügte Klasse:* Wird verwendet, um Fehlermeldungen, die vom Server gesendet werden zu interpretieren und in der korrekten Sprache auszugeben.

2.2 View

- **Login:** *hinzugefügt:* getUsername(), getServerAdress()
- **Lobby:** *hinzugefügt:* getChosenGameName(), getChatMessage(), hasPW-ChosenGame()
- **CreateWindow:** *entfernt:* addLeaveButtonListener(), addRulesetSelectionListener(); *hinzugefügt:* setRulesetTypes(), getGameName(), hasPassword(), getPassword(), getSelectedRulesetType()
- **GameLobby:** *hinzugefügt:* getChatMessage(), getSelectedPlayer()
- **Game:** *hinzugefügt:* getChatMessage(), addChatMessageListener() *geändert:* makeTrickGameBoard()
- **TrumpColour:** *Neu hinzugefügte Klasse:* Hinzugefügt um die aktuelle Trumpffarbe anzuzeigen. Das ist primär bei Wizard wichtig, da der Zauberer selbst keine Farbe hat und dementsprechend nicht genutzt werden kann, um die Trumpffarbe anzuzeigen.
- **Sonstiges:** Weiterhin wurden auch viele Detailänderungen an den einzelnen GamePanel Bausteinen vorgenommen.

2.3 Ruleset

- *hinzugefügt* **DiscardedCard:** Eine Karte die von einem Spieler auf den Ablegestapel gelegt wird und als Attribute die Karte die gespielt wurde und den Spieler der sie gespielt hat enthält.;
- *hinzugefügt* **EmptyCard:** Eine leere Karte die zu keinem Spiel gehört;

- **GameState:** *hinzugefügt:* madeTrick() Entfernt die DiscardedCards im Ablagestapel und gibt ihre Karten als Set einem Spieler in seine gemachten Stiche, restartDeck () Wird aufgerufen um ein neues Deck zu erstellen und alle Karten die vorher im Spiel waren zu löschen., nextRound() Erhöht die Rundenzahl um eins; *gelöscht:* getNumberOfPlayedCards()
- **ServerRuleset** *hinzugefügt:* updatePlayers() Schickt an alle Spieler ein neues GameClientUpdate, getPlayedCards() Holt die gespielten Karten im Ablagestapel, startRound() Wird am Anfang jeder Runde aufgerufen. Mischt und verteilt Karten und schickt Comobject mit Updates und Request für den Rundenanfang., getWinners() Wird bei Spielende aufgerufen und gibt die Namen der Sieger zurück,;
- **ServerWizard** *hinzugefügt:* trumpColour Die Trumpffarbe im Spiel;
- **ServerHearts:** *hinzugefügt:* heartBroken: Gibt an ob Herz gebrochen wurde. swapLeft(),swapRight(),swapAcross(): Tauscht die Karten nach links, rechts oder gegenüber

2.4 Server

```
/**
LoginServer. Diese Klasse ist für das Zustandekommen von
Clientverbindungen zustandig. Sie hat den ClientListenerTread
als innere Klasse.
/
```

- **LoginServer:** *Neu hinzugefügte Klasse, erbt von Server.*
Attribute:

```
/**
Ein Thread, der fuer das Annehmen neuer Clientverbindungen zustandig
ist
/
Thread clientListenerThread
/**
Der LobbyServer
/
LobbyServer lobby
```


Methoden:

```
/**
Diese ueberladene Methode ueberprueft, ob der Name im Set names des
LobbyServers vorhanden ist, falls ja, wird ein ComWarning an den Client
geschickt, dass der Name bereits vergeben ist, falls nein, wird im Player
setName aufgerufen. Der Player wecheselt auf den LobbyServer, sein Na-
```

me
wird in das names Set eingefügt. Dem Client wird ein ComInitLobby geschickt.

@param player
ist der Thread der die Nachricht erhalten hat
@param login
ist das ComObject, dass den Benutzernamen des Clients enthält
/
receiveMessage(Player player, ComLoginRequest login)
/**
Diese Methode schliesst die Verbindung zu einem Client. Der uebergebene Player wird aus dem playerSet sowie dem names Set im LobbyServer entfernt.

@param player
ist der Spieler der entfernt wird
/
disconnectPlayer()

- **GameServer:** *hinzugefügt:*

/**
Diese Methode beendet das Spiel und gibt die Player an den LobbyServer zurueck .Der GameServer wird aus dem Set des LobbyServers entfernt und die
Spieler in der Lobby erhalten ein Gamelist Update.

/
quitGame()
/**
Getter fuer den Namen des Spielleiters

@return Gibt den gameMasterName zurueck
/
getGameMasterName()
/**
Getter fuer das Passwort

@return Gibt das Passwort des Spieles zurueck
/
getPassword()

```

/**
Diese Methode verpackt eine RulesetMessage in ein ComObject und
verschickt es mit sendToPlayer() an einen bestimmten Spieler.
@param player
ist der Name des Spielers an den die Nachricht verschickt wird
@param message
ist die Ruleset Nachricht, die in ein ComObject verpackt wird
/
sendRulesetMessage(String player, RulesetMessage message)
/**
Diese Methode verschickt ComWarning mit der übergebenen Warnung
mit broadcast() an alle Spieler.
@param message
ist die Ruleset Nachricht, die in ein ComObject verpackt wird
/
broadcastRulesetMessage(RulesetMessage message)
/**
Diese Methode verschickt ein ComWarning mit sendToPlayer() an einen
bestimmten Spieler.
@param player
ist der Name des Spielers an den die Nachricht verschickt wird
@param message
ist die Ruleset Nachricht, die in ein ComObject verpackt wird
/
sendWarning(String player, WarningMsg warning)
/**
Diese Methode verpackt eine RulesetMessage in ein ComObject und
verschickt es mit broadcast() an alle Spieler.
@param message
ist die Ruleset Nachricht, die in ein ComObject verpackt wird
/
broadcastWarning(WarningMsg warning)
/**
Gibt zurück, ob das Spiel bereits gestartet ist
@return den Wert von hasStarted
/
isHasStarted()
/**
Setter für has started
@param hasStarted wird auf den übergebenen Wert gesetzt
/
setHasStarted(boolean hasStarted)
entfernt:
receiveMessage(Player player, ComClientQuit quit)–wird jetzt mit recei-
veMessage(Player player, ComClientLeave leave) gelöst

```

- **GameServerRepresentation:** *implementiert jetzt Serializable*
- **LobbyServer:** *entfernt:* innere Klasse ClientListenerThread— ist jetzt im Login Server
 Set|Player| noNames— wurde nicht gebraucht bzw. wurde durch den LoginServer einfacher gelöst
 ServerSocket socket— ist jetzt im Login Server
 receiveMessage(Player player, ComClientQuit quit)— macht der allgemeine Server
 receiveMessage(Player player, ComLoginRequest login)— ist jetzt im Login Server
hinzugefügt:
 /**
 Hilfsmethode, die einen Spieler zu einem Spiel hinzufuegt

 @param player ist der Spieler, der hinzugefügt wird
 @param game ist das Spiel zu dem der Spieler hinzugefügt wird
 /
 addPlayerToGame(Player player, GameServer game)
 /**
 Getter für das Set der Spielernamen

 @return Gibt das names Set zurück
 /
 getNames()
- **ClientListenerThread:** erbt jetzt von Thread *hinzugefügt:*
 /**
 Zeigt, ob der Tread auf verbindungen wartet
 /
 boolean waiting
 ServerSocket socket
 LoginServer server
- **Player:** erbt jetzt von Thread *hinzugefügt:*
 /**
 Socket des Clients
 /
 Socket connection

```

/**
Zeigt an, ob der Thread läuft
/
boolean run
/**
Diese Methode schließt den Socket, sowie comIn und comOut
/
closeConnection()

geändert: getName() heißt getPlayerName(), setName() heißt setPlayer-
Name()

```

- **Server:** *hinzugefügt:* eine receiveMessage Methode für jedes ComObject, das vom Server empfangen werden soll;
geändert: handleIOException heißt jetzt disconnectPlayer()
- **ServerMain:** *hinzugefügt:*
Attribut LoginServer loginServer

2.5 ComObjects

- **ComBeenKicked:** *gelöscht*, da das ComObject nicht mehr gebraucht wird.
- **ComGameEnd:** *gelöscht*, da das ComObject nicht mehr gebraucht wird.
- **ComServerAcknowledgement:** *gelöscht*, da das ComObject nicht mehr gebraucht wird.

3 Vergleich: Implementierungsplan und Realität

3.1 Änderungen am Plan

- Zuständigkeit von Arbeitspaket Ruleset(Wizard-Client) wurde von Patrick/Andi and Daniel gegeben, um die Arbeitszeiten anzugleichen

3.2 Milestone 1

Bereich	angen. Dauer	tats. Dauer	Abweichung	Begründung
View(Login+Lobby)	8 Stunden	4 Stunden	50%	(1)
View(Create+Join)	8 Stunden	5 Stunden	37,5%	(1)
View(GameLobby)	8 Stunden	3 Stunden	62,6%	(1)
Client(Login)	8 Stunden	8,5 Stunden	(-) 6,25%	(2)
Client(Lobby)	8 Stunden	6,5 Stunden	18,75%	
Client(Create+Join)	8 Stunden	6 Stunden	25%	
Server(Login)	16 Stunden	6 Stunden	62,5%	(5)
Server(Lobby)	8 Stunden	3 Stunden	62,5%	(5)
Server(Create+Join)	12 Stunden	6 Stunden	50%	(5)
Ruleset(Daten)	20 Stunden	6 Stunden	25%	(4)
Ruleset(ServerWizard)	30 Stunden	24 Stunden	33,4%	

- (1) Es waren noch Code-Teile aus der Pflichtenheft-Phase vorhanden.
- (2) Login wegen Änderungen am Loginserver umgebaut.
- (3) Viele der ComObjects waren bereits fertig.
- (4) Wurde hauptsächlich schon in Spezifikationsphase implementiert.
- (5) Stundenzahl würde Anfangs verdoppelt.

3.3 Milestone 2

Bereich	angen. Dauer	tats. Dauer	Abweichung	Begründung
View(Game)	20 Stunden	18,5 Stunden	7,5%	
Client (GameLobby)	8 Stunden	10 Stunden	(-) 25%	
Server(GameLobby)	10 Stunden	5 Stunden	50%	(1)
Ruleset(ServerHearts)	30 Stunden	18 Stunden	40%	
Handbuch	-	5 Stunden	-	

- (1) Stundenzahl würde Anfangs verdoppelt.

3.4 Milestone 3

Bereich	angen. Dauer	tats. Dauer	Abweichung	Begründung
View(WizardWindows)	4 Stunden	3,5 Stunden	12,5%	
View(HeartsWindows)	4 Stunden	3 Stunden	25%	
Client (Game)	14 Stunden	12 Stunden	14,3%	(1)
Ruleset(ClientWizard)	6 Stunden	5 Stunden	16,7%	
Ruleset(ClientHearts)	6 Stunden	5 Stunden	16,7%	
Server(Game)	4 Stunden	2,5 Stunden	37,5%	
Client(LanguageInterpreter)	-	2 Stunden	-	
Karten	-	4 Stunden	-	
Handbuch	-	2 Stunden	-	

- (1) Regelwerk Pakete delegiert. Zeit für Refactoring des Clientcodes verwendet.

3.5 Finale Phase

Bereich	angen. Dauer	tats. Dauer	Abweichung	Begründung
View(Polishing)	10 Stunden	7,5 Stunden	25%	
Client(Polishing)	10 Stunden	12 Stunden	(-) 20%	(1)
Ruleset(Polishing)	10 Stunden	5 Stunden	50%	
Server(Polishing)	10 Stunden	1 Stunde	10%	
Handbuch	-	3 Stunden	-	

- (1) Nacharbeiten wegen Nebenläufigkeit und JUnit Tests sowie Tests in Gruppe wegen Abgabe

4 Vergleich: Implementierung und Pflichtenheft

Kriterium	umgesetzt	Begründung
Text in Client können sich verbinden und eindeutigen Benutzernamen auswählen	ja	-
Lobby mit der Möglichkeit, Spiele zu erstellen und offenen Spielen beizutreten	ja	-
Unterstützung mehrere parallel laufender Spiele	ja	-
Regelauswertung mit Überprüfung erlaubter Aktionen, Punktezählung, Kartenausgabe	ja	-
Unterstützung der Spiele Hearts und Wizard	ja	-
Chat in der Lobby	ja	-
Chat mit Mitspielern während eines Spiels	ja	-
Schutz vor Cheats	ja	-
Darstellungsfenster, das mindestens bei der Auflösung von 1024x768 Bildpunkten benutzt werden kann	ja	-
Darstellung des laufenden Spiels (eigene Hand, verdeckte Hand der anderen Spieler, Ablage- und Aufnahmestapel, Punktestand, evtl. Zusatzinformationen)	ja	-
Eingabemöglichkeiten für erlaubte Aktionen während des Spiels (Karte ablegen, Ansagen, etc.)	ja	-
Die GUI muss den Benutzer sinnvoll unterstützen und benutzerfreundliche Eingabeelemente anbieten	ja	-
Flüssige Darstellung	ja	-
Unabhängigkeit vom Regelwerk	ja	-
Beim Start Auswahl des Servers und des Benutzernamens	ja	-
Anzeige von offenen Spielen in der Lobby mit Möglichkeit zum Erstellen und Beitreten	ja	-
Chat in der Lobby und während des Spiels	ja	-
Verwaltung der Verbindung mit dem Server	ja	-
Verwaltung des aktuellen Spielzustands (soweit Client bekannt)	ja	-
Vorab-Regelauswertung zur Unterstützung des Nutzers (ungültige Spielaktionen sind nicht durchführbar in der GUI)	ja	-
Passwortbeschränkung	ja	-
Statistiken	ja	-
Spielename	ja	-
Mehrsprachen	ja	-
Veränderbare GUI (Farben etc)	ja	-
Neue Runde direkt starten	nein	(1)

- (1) Es wurde auf andere Kriterien größerer Wert gelegt, z.B. Korrektheit des Spielablaufes.

5 Bekannte Bugs

- **Hearts - Spielen der Karten:** Momentan 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. Momentane Lösung: die isValidMove in ServerRuleset und ClientRuleset enthalten unterschiedlichen Code, so dass die Abfrage nach Herz nur vom Server gemacht wird. Die Funktion, dass jemand als Cheater vom Server erkannt wird, wenn diese Methoden sich unterscheiden, ist momentan unterbunden.
- **Abgelegte Karten:** Momentan verschwinden die abgelegten Karten sofort nachdem der letzte Spieler eine Karte gespielt hat. Dadurch ist es kaum möglich den Spielverlauf nachzuvollziehen. Die Karten sollen noch einige Sekunden angezeigt werden, bevor ein neuer Umlauf beginnt.
- **Heartspunkte:** Bei Hearts wird noch die Anzahl der gemachten Stiche statt der Punkte angezeigt.
- **GUI Hearts:** In Hearts kann man einige Komponenten, wie zb. die Anzeige der Trumpffarbe und -karte weglassen.
- **Fehlermeldungen:** Einige Fehlermeldungen können noch optimiert werden.
- **Kartenrückseite unter Linux:** Bild der Kartenrückseite wird unter Linux nicht sofort initialisiert.
- **Spielende und Scorefenster:** Das Spiel wird an Ende einfach beendet, eine Scorefenster wird nicht angezeigt.
- **GUI Wizard:** In der letzten Runde sollte keine Trumpfkarte mehr angezeigt werden.