

Pflichtenheft

Usman Ghani Ahmed
Philip Caroli
Maximilian Madlung
Jeremias Mechler
Fabian Neundorf

Praxis der Softwareentwicklung Gruppe 3

Entwicklung eines „Monopoly“-ähnlichen Spiels



WS 2010 / 2011

Inhaltsverzeichnis

1	Produktübersicht	4
1.1	Spielregeln	4
2	Zielbestimmung	5
2.1	Musskriterien	5
2.2	Sollkriterien	5
2.3	Kannkriterien	5
2.4	Abgrenzungskriterien	5
3	Produkteinsatz	6
3.1	Anwendungsbereiche	6
3.2	Zielgruppen	6
3.3	Betriebsbedingungen	6
4	Produktumgebung	6
4.1	Software	6
4.2	Hardware	6
4.3	Orgware	6
4.4	Schnittstellen	7
5	Funktionale Anforderungen	7
5.1	Client	7
5.1.1	Außerhalb des Spielfeldes	7
5.1.2	Bei hergestellter Verbindung	8
5.2	Server	10
5.2.1	Vorbereitungen	10
5.2.2	Spielzug	11
5.3	KI Client	12
5.3.1	Vorbereitungen	12
5.3.2	Spielzug	12
6	Produktdaten	12
6.1	Allgemein	12
6.2	Server	12
6.3	Client	13
6.4	KI-Client	13
7	Produktleistungen	13
8	Weitere nichtfunktionale Anforderungen	14
9	Qualitätsanforderungen	14

10 Globale Testfälle und Testszenarien	15
11 Systemmodelle	17
11.1 Übersicht	17
12 Benutzungsoberfläche	18
12.1 Hauptmenü	18
12.2 Warteraum	19
12.3 Spielfeld	20
13 Spezielle Anforderungen an die Entwicklungsumgebung	20
14 Zeit- und Ressourcenplanung	21
14.1 Projekt-Untergliederung	21
14.2 Ressourcen	21
15 Glossar	21

1 Produktübersicht

Das Produkt ist eine Umsetzung des berühmten Brettspielklassikers „Monopoly“ für den Computer. Ziel des Spiels ist es, durch Geschicklichkeit, strategisches Denken und der nötigen Portion Glück das ganze Geld der Gegner einzunehmen. Gewonnen hat der Spieler, der zuletzt das meiste Geld besitzt.

Das Spiel beinhaltet mehrere Spiel-Modi:

- Einzelspieler gegen Computer-KI
- Mehrspieler an einem oder mehreren Computern sowie mit als auch ohne KI

Das Programm zeichnet sich durch seine Benutzerfreundlichkeit aus, die sich in der intuitiven Bedienbarkeit der graphischen Benutzeroberfläche äußert. Abgebildet werden das Monopoly®-Spielfeld sowie einige wenige zusätzliche Elemente, um die Übersichtlichkeit zu wahren.

1.1 Spielregeln

Es können standardmäßig mindestens zwei und maximal acht Spieler an einem Spiel teilnehmen. Zu Beginn des Spieles bekommt jeder Spieler ein festgelegtes Startkapital, mit dem er Straßen, Häuser und Hotels kaufen kann oder Strafen sowie Steuerzahlungen die durch Ereignisse entstehen an die Bank bzw. in die Mitte des Spielbretts bezahlen muss.

Auf einer Straße kann man zusätzlich noch Hotels und Häuser bauen, dadurch erhöht sich die Miete der Straße. Das Geld, das sich auf dem Spielbrett sammelt, bekommt derjenige Spieler, der auf dem Feld „Frei parken“ zum Stehen kommt. Es treten verschiedene Spieler gegeneinander an, dabei können die Spieler menschlich sein oder werden vom Computer (KI) gesteuert.

Jeder Spieler kommt abwechselnd und zyklisch an die Reihe und würfelt mit zwei Würfel. Die Summe, die sich aus den Augenzahlen der zwei Würfel ergibt, gibt dem Spieler Auskunft darüber, wie viele Schritte er mit seiner symbolischen Figur voranschreiten darf. Die Spielfiguren bewegen sich im Uhrzeigersinn auf dem Spielfeld. Dabei überquert der Spieler im Laufe des Spiels verschiedene Straßen, die von Spielern für einen festgelegten Preis gekauft werden können. Bleibt der Spieler auf einer Straße, die einem anderen Spieler gehört, stehen, so muss er einen gewissen Betrag an Geld an diesen Spieler zahlen (Miete). Die Höhe des Einnahmegeldes (Miete) hängt davon ab, ob der Spieler nur diese Straße besitzt oder ob er auf dieser Straße noch weitere Gebäude (Häuser, Hotels) gebaut bzw. gekauft hat. Wenn der Spieler zusätzlich noch Häuser und Hotels auf dieser Straße gebaut hat, dann erhöht sich das Einnahmegeld für die jeweilige Straße. Dabei steigt die Miete in aufsteigender Reihenfolge, d.h die Miete für ein Hotel ist höher als die für ein Haus.

Landet der Spieler auf einem „Steuern bezahlen“-Feld, dann muss er Steuern bezahlen, die sich in der Mitte des Spielbretts sammeln. Bleibt der Spieler auf dem Feld „Gefängnis“ stehen, so setzt er eine Würfelrunde aus, kann jedoch immer noch Miete von den anderen Spielern einnehmen, sowie Handel mit anderen Spielern betreiben. Überquert ein Spieler das Feld „Los“, auf welchem zu Beginn gestartet wurde, so kassiert er eine Prämie. Beim Stillstand auf einem Ereignisfeld zieht der Spieler eine Ereigniskarte mit verschiedenen Aufgaben, die der Spieler erledigen muss und dabei auch Geldprämien kassieren kann oder Abgaben bezahlen muss. Jeder Spieler hat während des Spiels die Möglichkeit, seine Gebäude und Straßen an die anderen Spieler zu verkaufen, sowie diese für eine Ersteigerung (Auktion) freizugeben. Kann ein Spieler finanziell seinen Forderungen nicht mehr nachkommen, so scheidet er automatisch aus. Spieler, die vorzeitig ein laufendes Spiel beenden, können ihre Habe entweder an die Bank abgeben oder an ihre Mitspieler. Gewonnen hat der Spieler, der zuletzt das meiste Geld besitzt. Ziel dieses Spieles ist, durch Geschicklichkeit und ein bisschen Glück seine Gegner finanziell in die Knie zu zwingen.

2 Zielbestimmung

2.1 Musskriterien

- Plattformunabhängigkeit (Java, Swing)
- Umsetzung der Monopoly-Regeln
- Client-Server-Architektur
- Spielen gegen KI-Client mit Fähigkeitslevel eines Monopoly-Anfängers möglich
- Anzeigen und Interaktion mit dem Spielfeld in Form einer grafischen Benutzeroberfläche
- Spielen mit variabler Anzahl echter Spieler und KI-Spieler auf einem Server gegeneinander möglich

2.2 Sollkriterien

- Spielplan, Text und Grafik sind variabel
- Benutzeroberfläche sieht dem Monopoly-Brett ähnlich
- Netzwerkfähigkeit

2.3 Kannkriterien

- Spielregeln lassen sich variieren
- Lokalisierbarkeit
- Text-Chat
- Zuschauermodus
- Server-Admin kann Spiel verwalten
- Unterbrechen, Speichern und Fortsetzen eines Spiels
- Hilfe

2.4 Abgrenzungskriterien

- Umsetzung für mobile Systeme
- Finden von Internetservern (Metaserver)
- Voice-Chat
- verschiedene KI-Stufen

3 Produkteinsatz

3.1 Anwendungsbereiche

- Privater PC, Freizeit

3.2 Zielgruppen

- Menschen mit Spaß an Gesellschaftsspielen aus dem deutschsprachigen Raum

3.3 Betriebsbedingungen

- Dauer eines Spiels: Circa ein bis drei Stunden

4 Produktumgebung

4.1 Software

- Java Runtime Environment SE 1.6 oder neuer
- Betriebssystem zum Beispiel Windows, Linux, OS X

4.2 Hardware

Mindestanforderungen (PC):

- 128 MB Arbeitsspeicher
- 100 MB freien Festplattenspeicher
- 500-MHz-Prozessor
- Bildschirmauflösung: 1024×768
- Beschleunigte 2D-Grafik

Empfohlen:

- 512 MB Arbeitsspeicher
- 100 MB freien Festplattenspeicher
- 1-GHz-Prozessor
- Bildschirmauflösung: 1280×1024
- Beschleunigte 2D-Grafik
- Netzwerk- oder Internetverbindung (TCP/IP, IPv4)

4.3 Orgware

entfällt

4.4 Schnittstellen

- Vorgegebene Client-Server-Schnittstelle (Java-Interface)
- Netzwerk-Client-Server-Schnittstelle

5 Funktionale Anforderungen

Jede funktionale Anforderung wird durch eine fünfstellige Nummer gekennzeichnet. Die erste Stelle gibt die Anwendung an:

1. Client
2. Server
3. KI-Client

Die zweite Stelle gibt den Bereich innerhalb des Spielverlaufs an.

1. Vorbereitungen
2. Spielzug

Die drei letzten Stellen werden dann durchnummeriert.

5.1 Client

5.1.1 Außerhalb des Spielfeldes

/F11010/ Anzeigen des Hauptmenüs

Beim Starten des Programms soll das Hauptmenü angezeigt werden.

Im Hauptmenü können folgende Punkt ausgewählt:

- Einstellungen (→ /F11020/)
- Spiel erstellen (→ /F11050/)
- Spiel beitreten (→ /F11100/)
- Spiel speichern (→ /F11700/) (nur bei aktivem Spiel)
- Beenden

/F11020/ Ändern der Spieleinstellungen

Zuerst werden die Einstellungen geladen und in das Fenster übertragen (→ /F11030/). Danach kann man diese ändern und können dann gespeichert werden (→ /F11040/).

/F11030/ Laden der Einstellungen

Die in den Einstellungen (→ /F11020/) vorhandenen Daten werden vor dem Anzeigen aus der Datei geladen.

/F11040/ Speichern der Einstellungen

Die im Einstellungsfenster (→ /F11020/) eingestellten Einstellungen werden gespeichert.

/F11050/ Erstellen eines Servers (Konfiguration)

Es ist möglich einen Server zu starten. Dazu kann man eine Konfiguration laden (→ /F11060/), ändern und speichern (→ /F11070/). Alternativ kann auch ein Spielstand geladen werden (→ /F11160/). Danach kann der Server dann gestartet werden.

/F11060/ Laden der Einstellungen

Die im Konfigurationsfenster des Servers (→ /F11050/) angezeigten Einstellungen werden vor dem Anzeigen aus der Datei (→ /DB02/) geladen.

/F11070/ Speichern der Einstellungen

Die in den Einstellungsfenster (→ /F11050/) eingestellten Einstellungen werden gespeichert.

/F11080/ Erstellen eines Servers

Nachdem die Einstellungen durchgeführt wurden (→ /F11050/) kann ein Server gestartet werden, dadurch wird (→ /F11090/) aufgerufen.

/F11090/ Starten eines Servers

Ein Server wird initialisiert und gestartet, er befindet sich dann im Bereitschaftsmodus und stellt den Spielern einen „Warteraum“ zur Verfügung.

/F11100/ Spiel beitreten

Im Hauptmenü (→ /F11010/) kann einem Spiel beigetreten werden, je nach Benutzerwahl können dann „Direkte Verbindung“ (→ /F11110/) oder die „Serverliste“ (→ /F11120/) ausgewählt werden. Danach wird ein eventuell vorhandenes Passwort und die Namen sowie die Anzahl der Spieler abgefragt und dann die Verbindung aufgebaut (→ /F12010/). Sofern das Spiel bereits läuft wird, der Vorgang abgebrochen.

/F11110/ Spiel beitreten durch Eingabe einer IP (direkte Verbindung)

Die IP wird angenommen und es wird getestet, ob mit der IP ein Server erreichbar ist und Plätze frei hat. Sind keine Plätze frei, wird damit abgebrochen.

/F11120/ Anzeigen der Serverliste

In dieser Liste können einige Server angezeigt werden, ohne dessen IP zu kennen (→ /F11130/).

In dieser Liste kann ein Server angewählt werden, sofern noch Plätze frei sind. Ansonsten ist eine Verbindung nicht möglich.

/F11130/ Ermitteln der Server im lokalen Netzwerk

Die im Netzwerk verfügbaren Server werden auf die Liste aufgenommen und die Serverinformationen werden abgefragt.

/F11140/ Abfragen der Serverinformationen für die Serverliste

Jeder Server in der Liste übersendet wichtige Informationen über sich, um sie dann in der Liste anzuzeigen (z. B. ob ein Passwort vorhanden ist, wie viele Spieler sich verbunden haben, wie viele Spieler sich verbinden dürfen).

/F11150/ Verlassen des Spiels

Über die entsprechende Schaltfläche im Programm kann das Spiel verlassen werden.

/F11160/ Spielstand laden

Sofern man ein Spielstand lädt wird die Konfiguration automatisch daraus geladen.

/F11170/ Spielstand speichern

Man kann im Hauptmenü den Spielstand jederzeit speichern.

5.1.2 Bei hergestellter Verbindung

/F12010/ Initiieren eines Spiels

Der Warteraum wird angezeigt. Währenddessen kann man sich mit den anderen Spielern unterhalten (→ /F12020/). Nachdem mindestens zwei Spieler verbunden sind und alle Spieler bereit sind, startet das Spiel (→ /F12030/).

/F12020/ Chatten

Es ist während des gesamten Spiels möglich, mit anderen Spielern zu chatten.

/F12030/ Starten des Spiels

Nachdem das Spiel gestartet ist, erscheint bei allen das Spielfeld, auf dem folgende Punkte angezeigt werden:

- Spieler mit deren Namen und Kontostände
- Die Straßen mit deren Zugehörigkeit und Bebauung
- Die Spielfiguren

- Interaktionspopup
 - Das Geld in der Mitte
 - Handel
 - Versteigerungen
 - Ereigniskarten
- Das Chatfenster

Daraufhin kann er der erste Spieler die Runde starten (\rightarrow /F12040/). Außerdem kann jeder während des gesamten Spiels Nachrichten versenden (\rightarrow /F12020/).

/F12040/ Durchführung einer Runde

Sobald der nächste Spieler aktiv ist, darf dieser selber würfeln und somit den Zug beginnen (\rightarrow /F12050/).

/F12050/ Würfeln

Zum Würfeln fragt der aktive Spieler eine Zufallszahl beim Server ab (\rightarrow /F22020/). Anhand dieser Zufallszahl ziehen die Spieler die Spielfigur des aktiven Spielers weiter.

Landet der Spieler auf einen Sonderfeld, wird entsprechend dem Sonderfeld vorgegangen. Danach kann der aktive Spieler weitere Aktionen durchführen (\rightarrow /F12060/).

/F12060/ Weitere Aktionen

Nachdem der Spieler gewürfelt hat, kann er folgende Aktionen ausführen:

- den Spielzug beenden (\rightarrow /F22010/ /F12040/).
- eine Straße kaufen (\rightarrow /F22031/ /F12061/).
- ein Gebäude bauen (\rightarrow /F22032/ /F12062/).
- eine Hypothek aufnehmen (\rightarrow /F22033/ /F12063/).
- ein Gebäude abreißen (\rightarrow /F22034/ /F12064/).
- einen Handel initiieren (\rightarrow /F22035/ /F12065/).
- eine Straße versteigern (\rightarrow /F22036/ /F12066/).

Außerdem kann währenddessen die Miete vom Besitzer der Straße eingefordert werden (\rightarrow /F12070/).

/F12061/ Straße kaufen

Da der Spieler nur die Straße kaufen kann, auf den der Spieler steht, wird vorher nur geprüft, ob die Straße zu verkaufen ist und der Spieler genügend Geld hat (\rightarrow /F22031/). Danach kann der Spieler weitere Befehle ausführen (\rightarrow /F120Y0/).

/F12062/ Gebäude bauen

Zuerst wird geprüft, ob der Spieler genügend Geld hat und nach den entsprechenden Regeln dort ein Gebäude gebaut werden darf. (\rightarrow /F22032/)

/F12063/ Hypothek aufnehmen

Wenn der Spieler nach den Regeln eine Hypothek aufnehmen darf, meldet der aktive Spieler dies dem Server (\rightarrow /F22033/).

/F12064/ Gebäude abreißen

Erst wird überprüft, ob der Spieler das Gebäude nach den Spielregeln abreißen darf (\rightarrow /F22034/).

/F12065/ Handel initiieren

Der aktive Spieler kann ein Handel mit einem anderen Spieler oder der Bank initiieren (\rightarrow /F22035/). Darauf hin können beide Spieler ihr Angebot einstellen und dieses fixieren. Danach müssen beide dann das Angebot bestätigen und der Handel erfolgt.

/F12066/ Straße versteigern

Um eine Straße zu versteigern, müssen dies die Regeln erlauben. Danach kann jeder Spieler darauf bieten (\rightarrow /F22036/).

/F12070/ Miete erheben

Sobald ein Spieler auf ein Feld zieht, das ihm nicht gehört, kann der Besitzer die Miete erheben. Dies kann er solange, bis der nächste Spieler gewürfelt hat.

Nachdem der Spieler fertig ist, kann er die aktuelle Runde an den nächsten Spieler abgeben. Außerdem kann ein Spieler aufgeben (\rightarrow /F20/).

/F12080/ Aufgeben

Jeder Spieler kann, sobald er an der Reihe ist, aufgeben. Er kann dann sagen, was mit seinem verbleibenden Besitz passiert:

- Die Bank bekommt das Geld, die Ereigniskarten und die Straßen zurück. Je nach Einstellungen bleiben die Gebäude stehen oder werden abgerissen.
- Zuerst werden alle Straßen an andere Spieler versteigert, dann bekommt die Bank die übrig gebliebenen Straßen, das Geld und die Ereigniskarten. Wieder können je nach Einstellungen die Gebäude stehen bleiben.
- Es ist möglich, das Eigentum an andere Spieler zu verschenken. Alles Übriggebliebene geht an die Bank. Eventuell vorhandene Straßen mit Gebäuden behalten die Gebäude abhängig von den Einstellungen.

5.2 Server

5.2.1 Vorbereitungen

/F21010/ Starten des Servers

Bei den Starten des Servers wird ermittelt, ob ein Spielstand geladen (\rightarrow /F21020/) oder ein neues Spiel gestartet werden soll (\rightarrow /F21030/).

/F21020/ Spielstand laden

Sofern der Spielstand valide ist, wird diese Datei eingelesen und der Server ist im Wartemodus (\rightarrow /F21040/).

/F21030/ Neues Spiel starten

Es wird ein neues Spiel gestartet, indem die entsprechenden Anfangseinstellungen aus den Übergabeparametern oder einer Datei geladen werden. Danach befindet sich der Server im Wartemodus (\rightarrow /F21040/).

/F21040/ Wartemodus

Im Wartemodus stellt der Server einen Chatraum zur Verfügung. Jeder Spieler landet zuerst hier und kann sich auf 'bereit' setzen. Nachdem mindestens zwei Spieler verbunden und alle bereit sind, startet das Spiel (\rightarrow /F21050/), sofern es sich um ein geladenes Spiel handelt. Ansonsten, wenn es sich um ein neues Spiel handelt, werden die Vorbereitungen abgeschlossen (\rightarrow /F21060/).

/F21050/ Spielstart

Nachdem allen Spielern die Reihenfolge übermittelt wurde, wird der erste Spielzug gestartet, indem der entsprechende Spieler informiert wird, dass er dran ist (\rightarrow /F22040/).

/F21060/ Vorbereitungen eines neuen Spiels abschließen

Wenn es sich um ein neues Spiel handelt, wird die Reihenfolge der Spieler festgelegt, und der erste Spieler in der neuen Reihenfolge ist dann der erste Spieler, der einen Zug ausführen darf. Danach startet das Spiel (\rightarrow /F21050/).

5.2.2 Spielzug

/F22010/ Spielzug informieren

Nachdem die Person bekannt ist, die nun zieht, wird allen Spielern übermittelt, dass dieser Spieler dran ist. Danach wartet der Server darauf, dass gewürfelt werden soll (\rightarrow /F22020/). Währenddessen stellt der Server einen Chatraum zur Verfügung. Überschreitet der aktive Spieler eine Bedenkdauer, wird der Spieler herausgeworfen und alles geht an die Bank.

/F22020/ Würfeln

Um den Betrug des Spielers vorzubeugen, bekommt der Server von den Client den Befehl zu würfeln. Dieser erzeugt dann eine zufällige Zahl zwischen 2 und 12. Alle Spieler erhalten dann diese Zahl und alle Spieler ziehen die Spielfigur des aktiven Spielers weiter. Danach kann der aktive Spieler weitere Befehle geben (\rightarrow /F22030/).

/F22030/ Weitere Befehle

Der Server bekommt jetzt die weiteren Befehle vom entsprechenden Client. Bei allen der folgenden Befehle wird anfangs getestet ob der Zug erlaubt ist:

- Der nächste Spieler kann an der Reihe sein (\rightarrow /F22010/).
- Der Spieler möchte die Straße kaufen (\rightarrow /F22031/).
- Der Spieler möchte ein Gebäude bauen (\rightarrow /F22032/).
- Der Spieler kann eine Hypothek aufnehmen (\rightarrow /F22033/).
- Der Spieler kann ein Gebäude abreißen (\rightarrow /F22034/).
- Der Spieler kann handeln (\rightarrow /F22035/).
- Der Spieler kann eine Straße versteigern (\rightarrow /F22036/).
- Der Spieler kann aussteigen (\rightarrow /F22037/).

Alle Spieler werden darüber nach erfolgreicher Ausführung informiert.

/F22031/ Kauf einer Straße

Der Kontostand des Spielers wird entsprechend des Kaufpreises belastet und als Besitzer der Straße der aktuelle Spieler eingetragen.

/F22032/ Bau eines Gebäudes

Der Kontostand wird mit den Gebäudepreis belastet und der Straße ein Gebäude hinzugefügt.

/F22033/ Aufnehmen einer Hypothek

Bei der Aufnahme der Hypothek wird der Kontostand des Spielers erhöht.

/F22034/ Gebäudeabriss

Dem Kontostand wird der entsprechende Abrisswert hinzugefügt und das Gebäude von der Straße entfernt.

/F22035/ Handel

Beim Handel wird der Handelspartner über den Beginn eines Handels informiert. Dann können beide Spieler ihre Angebote hinzufügen (\rightarrow /F22041/). Sofern ein Spieler fertig ist, kann dieser das Angebot fixieren und keine Änderungen an dem eigenen Angebot vornehmen. Sobald beide das als Angebot fixiert haben, können diese es annehmen und der Handel erfolgt.

/F22036/ Versteigern einer Straße

Zuerst wird getestet, ob die Spielregeln es erlauben, die Straße zu verkaufen. Sofern die Regeln es erlauben, wird die Straße den Personen zum Verkauf angeboten. Alle Spieler können dann darauf bieten und entsprechend den Preis erhöhen. Nach der Versteigerung werden alle Spieler über den neuen Besitzer der Straße informiert.

/F22037/ Ausstieg eines Spielers

Ein Spieler kann aussteigen und seinen Besitz entweder an die Bank zurückgeben, verkaufen oder versteigern.

/F22040/ Spielstand speichern

Es gibt einen Befehl um den Spielstand zu speichern.

5.3 KI Client

5.3.1 Vorbereitungen

/F31001/ Starten des KI-Client

Der KI-Client verbindet sich beim Start automatisch mit einem Server, dessen Beschreibung er per Befehlsparameter erhalten muss

/F31002/ Betreten eines Servers

Wenn der KI-Client einem Server beitrifft, setzt er sich automatisch auf „bereit“ und wartet auf den Beginn des Spiels

/F31003/ Laden der KI-Parameter

Der Client lädt die Parameter für seine KI-Logik aus einer angegebenen Datei

5.3.2 Spielzug

/F32001/ Durchführen eines Zuges

Der KI-Client würfelt automatisch und kann danach alle Spielaktionen ausführen, die auch der Spielerclient ausführen kann.

/F32002/ Ausführen einer Spielaktion

Die interne KI-Spiellogik entscheidet, ob und welche Aktion ausgeführt werden soll

/F32030/ Verlangen der Miete

Sobald ein anderer Spieler auf einem Spielfeld, das dem KI-Klienten gehört landet, verlangt dieser die anfallende Miete

/F32004/ Speichern der KI-Parameter

Sobald sich die KI-Parameter verändern, werden die neuen Parameter in der Datei gespeichert, aus der sie in /F31003/ geladen wurden

6 Produktdaten

6.1 Allgemein

Folgende Daten müssen sowohl Server als auch Client vorliegen:

/PD01/ Spielfeld

- Straßennamen
- Währungszeichen, Währungsnamen
- Beschriftung der Sonderfelder (“Frei parken” usw)
- Text der Ereigniskarten
- Name / Beschreibung der baubaren Objekte
- optional: Grafiken für die Spielfelder

6.2 Server

/PD02/ Servereinstellungen

- Servername
- Passwort (optional)
- Mindestspielerzahl
- Höchstspieleranzahl

/PD03/ Spielstand

- Liste aller teilnehmenden Spieler
- Kontostand jedes Spielers
- Besitz an Spielfeldern und Gebäuden jedes Spielers
- Hypotheken auf Spielfeldern
- aktueller Spieler
- Verteilung der Ereigniskarten
- Anzahl und Name der KI-Clients

6.3 Client

/PD04/ Client-Einstellungen

- Spielername
- IP des zuletzt beigetretenen Servers
- präferenzierte Größe des Fensters

6.4 KI-Client

/PD05/ KI-Einstellungen

- Name (optional)
- Parameter der KI-Logik

7 Produktleistungen

Zeit:

- Starten eines Servers: < 15 Sekunden
- Beitreten eines Servers: < 15 Sekunden
- Öffnen des Hauptmenüs: < 15 Sekunden
- Ausführen einer Aktion: < 2 Sekunden
- Erkennen eines Verbindungsabbruchs: < 30 Sekunden
- Beenden eines Spiels: < 10 Sekunden

Genauigkeit:

- Alle Aktionen sollen genau / fehlerfrei zwischen Server und Client übertragen werden

8 Weitere nichtfunktionale Anforderungen

Urheber- und Markenrechte:

- „Monopoly“ ist ein eingetragenes Warenzeichen, d.h. das Produkt darf nicht „Monopoly“ oder einen ähnlich klingenden / verwechselbaren Namen tragen
- Der Spielplan, das Regelwerk (als Textstück), die Figuren und alle Graphiken sind unter Schutzrecht, d.h. die sichtbaren Elemente des Produkts müssen von diesen Abweichen
- Spielregeln sind generell nicht schutzfähig (Patentgesetz und Gebrauchsmustergesetz 1.2), wobei der genaue Regelaufbau vielleicht trotzdem noch verändert werden sollte.

Sicherheitsanforderungen:

- Es darf nicht möglich sein, durch Server oder Client des Produktes andere Programme o.ä. zu beeinflussen
- Alle Spielaktionen sollten vom Server verifiziert werden, um „Cheaten“ zu verhindern

Plattformabhängigkeiten:

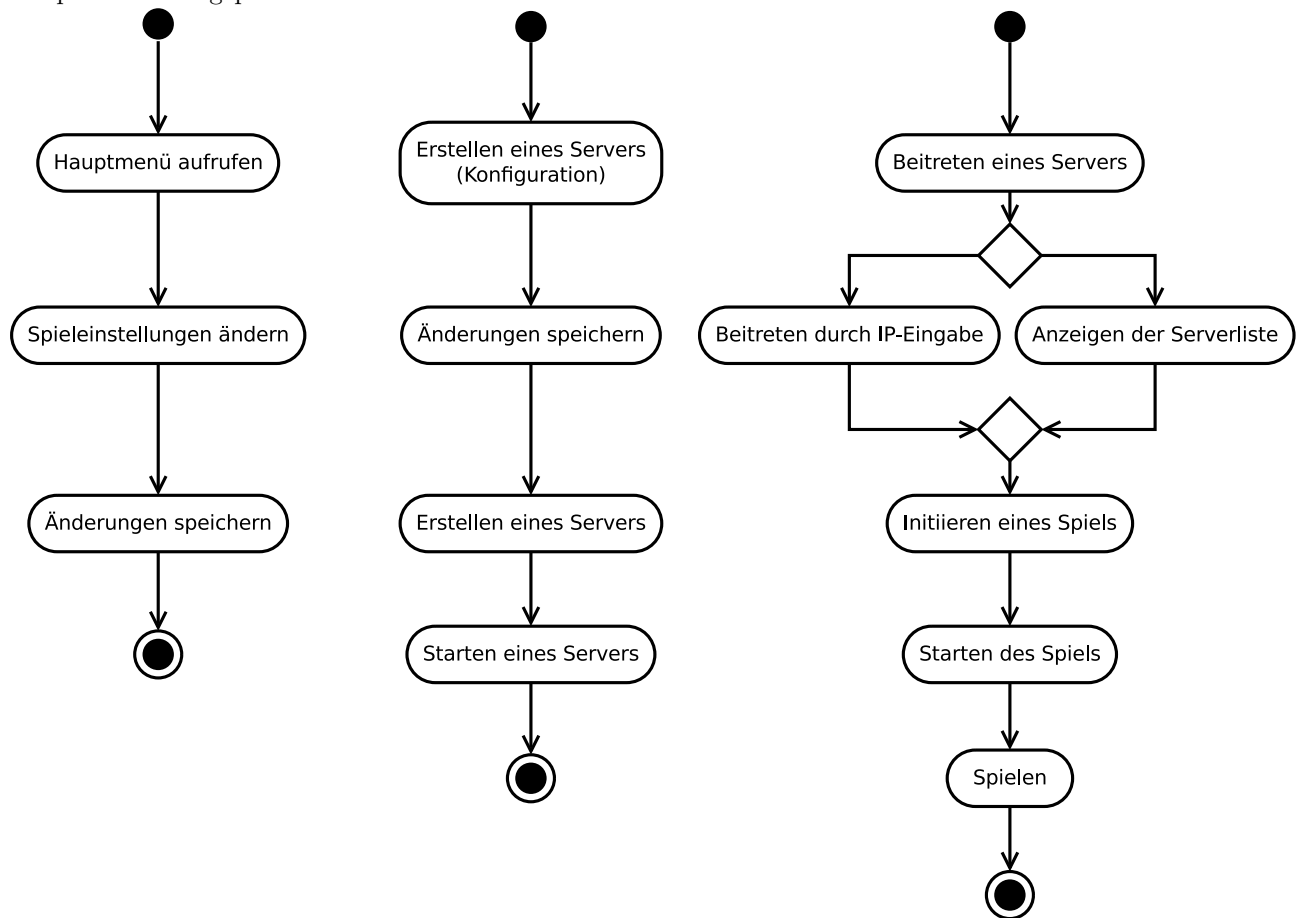
- Das Produkt muss auf der offiziellen Java-VM von Oracle laufen

9 Qualitätsanforderungen

- Das Produkt muss zuverlässig laufen und darf nicht ohne ausführliche Fehlermeldung crashen
- Es müssen durchschnittlich mindestens fünf Spiele in Folge gespielt werden können, ohne dass ein Fehler auftritt
- Das Programm verhält sich Verbindungsabbrüche betreffend robust und versucht bei Auftreten eines solchen das Spiel wiederherzustellen
- Es muss möglich sein, nach einem Fehler einem angefangen Spiel wieder beizutreten (Client) bzw. ein Spiel wieder so zu Starten wie es vor dem Fehler lief (Server)

10 Globale Testfälle und Testszenarien

Folgende Funktionssequenzen sind in der grafischen Benutzeroberfläche durchzuführen. Der Server wird entsprechend mitgeprüft:



Folgende Datenkonsistenzen müssen eingehalten werden:

- Zu jedem Zeitpunkt muss ein Spielername definiert sein
- Einstellungen können nur geladen werden, wenn die zu ladende Datei vollständig ist
- Einstellungen können nur gespeichert werden, wenn alle Felder korrekt ausgefüllt sind
- Der Zustand des Spielfeldes muss zwischen Client und Server identisch sein

Folgende unzulässigen Aktionen müssen korrent behandelt werden:

- Unzulässiges Übersenden von Kommandos an Client und Server (DoS)
- Überschreitung der maximalen Teilnehmeranzahl
- Belegung von Ressourcen durch häufiges Beitreten und Verlassen von Clients
- Unerwarteter Verbindungsabbruch (Timeout, Connection Reset)
- Reaktion auf Verbindungsverlust zum Server
- Unsauberer Restart des Clients
- Übersenden von undefinierten und unzulässigen Befehlen an Client und Server

Interoperabilität, Usability:

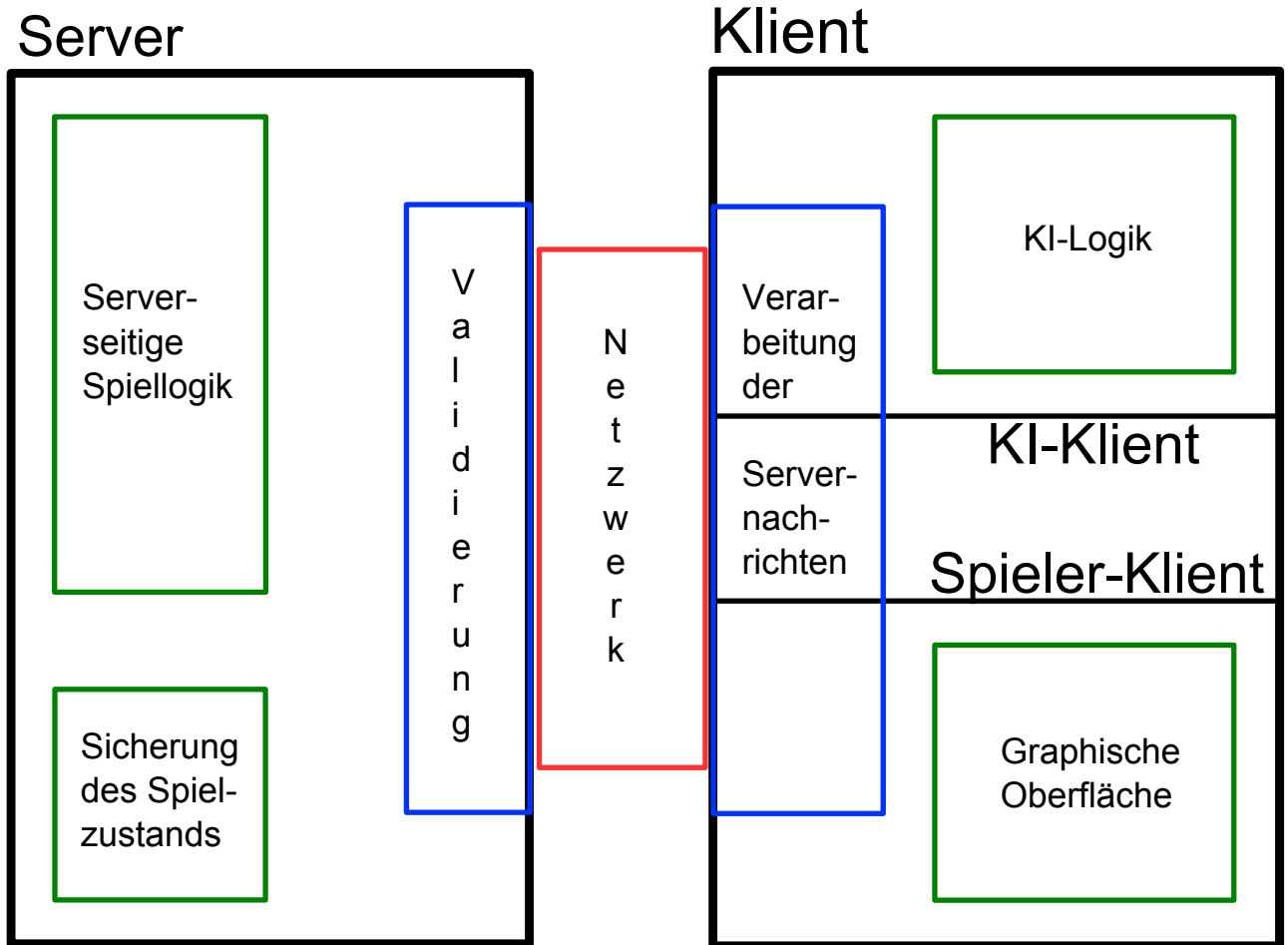
- Richtige Funktion mit Client und Server der anderen Gruppe
- Korrekte Darstellung auf verschiedenen Betriebssystemen (Windows, Linux, OS X)
- Verschiedene Bildschirmauflösungen
- nicht-standardmäßige Systemschriftgröße

Testszenarien:

- “Erste Inbetriebnahme mit Starten eines Servers”
Starten des Programms → im Hauptmenü „Einstellungen“ aufrufen → Daten eingeben → “Spieleinstellungen speichern” anklicken → im Hauptmenü “Spiel erstellen” anklicken → Einstellungen vornehmen → “Einstellungen speichern” anklicken → “Erstellen des Servers” anklicken → “Starten des Servers” anklicken.
- “Einem Spiel beitreten (durch Serverliste), spielen und aufgeben”
Starten des Programms → “Spiel beitreten” auswählen → „Serverliste“ anklicken → Server auswählen → verbinden → Bereitschaft setzen → Würfeln, Spielzüge durchführen → aufgeben
- „Einzelspieler“
“Spiel erstellen” anklicken → Mindest-Anzahl der Spieler > 1 setzen, andere Einstellungen vornehmen → “Einstellungen speichern” anklicken → “Erstellen des Servers” anklicken → “Starten des Servers” anklicken.
- “Voller Server”
Spieler versucht auf vollen Server zu verbinden → erhält Meldung, dass die maximale Teilnehmeranzahl erreicht ist
- „Serverabsturz“
Mehrere Spieler spielen → Server stürzt ab → Server wird neu gestartet → Spiel wird über Sicherungs-Datei auf dem Server wiederhergestellt
- “Abwesenheit eines Spielers”
Ein Spieler muss einen Zug vornehmen und reagiert eine bestimmte Zeit nicht → er wird zwangsweise aus dem Spiel entfernt → das Vermögen fällt der Bank zu

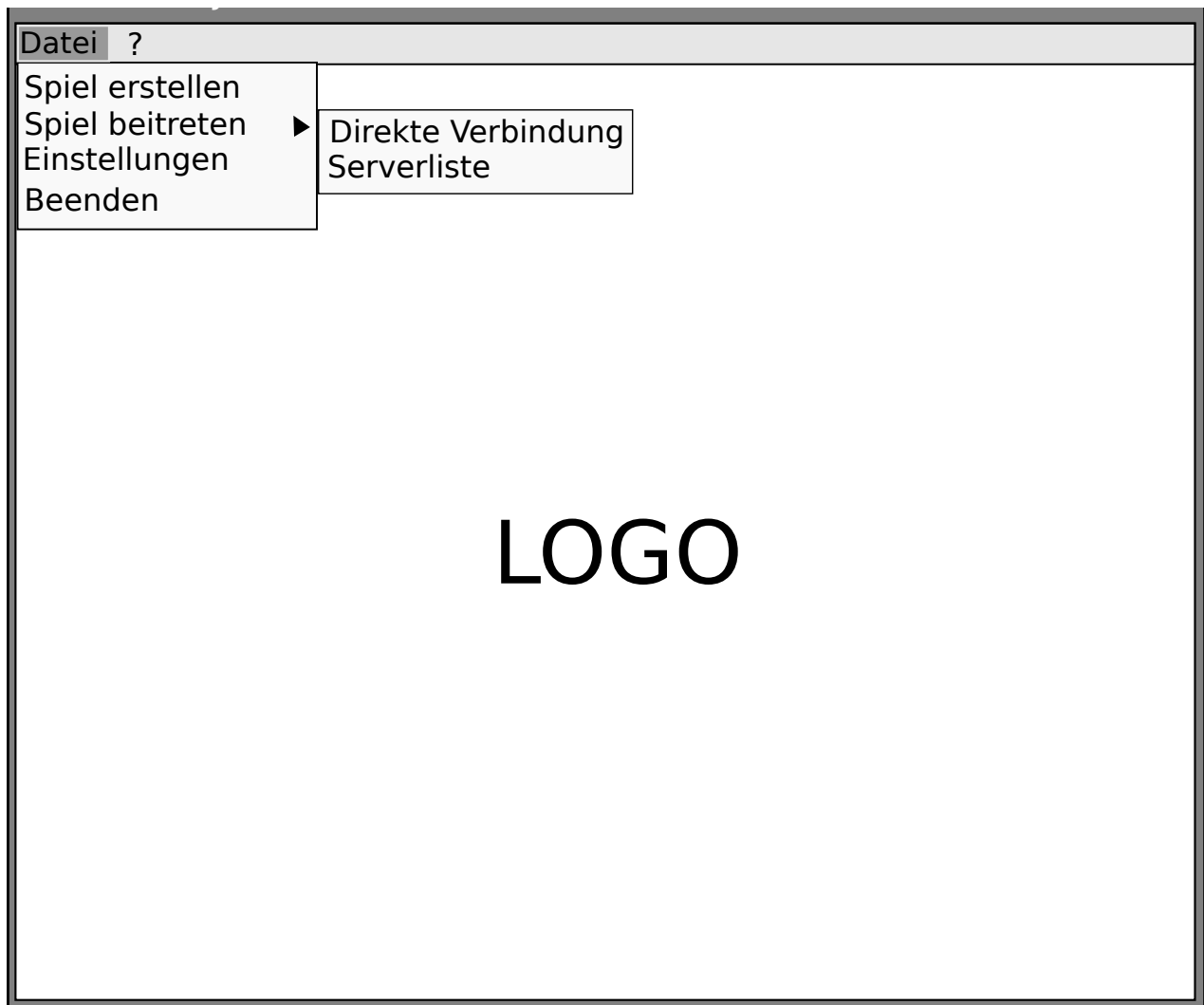
11 Systemmodelle

11.1 Übersicht



12 Benutzungsoberfläche

12.1 Hauptmenü



12.2 Warteraum

Datei ?

Vikath: Hallo
Usman: Ihr habt keine Chance!
Max: Klickt mal bitte auf Bereit!

Wo???

☐ xZise

☐ Vikath

☐ Max

☐ Jeremias

☐ Usman

☒ Bereit?

12.3 Spielfeld

Datei ?

Interaktionspopup

xZise 400€

Vikath 1000€

Max 0€

Jeremias 100€

Usman 10000€

Vikath: Hallo
Usman: Ihr habt keine Chance!

HAHA!

Kaufen

WÜRFELN

LOS!

500 €

1500 €

13 Spezielle Anforderungen an die Entwicklungsumgebung

- Allgemein
 - L^AT_EX
 - Versionskontrolle (SVN, Git)
- Entwicklung
 - IDE: Eclipse, Netbeans
- Entwurf
 - Geeignete UML-Tools wie Rational Architect, DIA
- Validierung
 - JUnit
- Teamkommunikation
 - Forum
 - E-Mail (Verteiler)

14 Zeit- und Ressourcenplanung

14.1 Projekt-Untergliederung

Zu planende und implementierende Module	Zeit	Personen
KI-Spiellogik	75h	Jeremias, Fabian
Client-Grundgerüst	40h	Fabian
GUI - Design + Funktionalität	30h	Philip, Max
Menüführung + Funktionalität Klient	20h	Usman
serverseitige Spiellogik	100h	Philip, Max
Sicherung des Spielzustands	20h	Usman
Verifizieren der Klient-Eingabe	10h	Jeremias
Netzwerkanbindung	20h	Usman

Präsentation des Pflichtenhefts: Philip Caroli

14.2 Ressourcen

- Rechner mit Leistungswerten in der Nähe der Mindestanforderungen
- Zwei oder mehr über Netzwerk verbundene Rechner

15 Glossar

- Metaserver (syn. Masterserver)
Eine mögliche Ergänzung. Speichert die IPs von mehreren Servern, sodass sie in der Serverliste angezeigt werden können.
- IDE Integrated Development Environment
Entwicklungstool, welches Editor, Compiler, Linker und Debugger enthält
- Client
Ein Client ist ein Programm, welches zu einem Server Kontakt aufbaut und mit diesem Daten austauscht.
- Server
Ein Server stellt Dienste für den Client bereit und tauscht Daten mit ihm aus.