

# 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

<b>1</b>	<b>Produktübersicht</b>	<b>4</b>
1.1	Spielregeln . . . . .	4
<b>2</b>	<b>Zielbestimmung</b>	<b>4</b>
2.1	Musskriterien . . . . .	4
2.2	Sollkriterien . . . . .	5
2.3	Kannkriterien . . . . .	5
2.4	Abgrenzungskriterien . . . . .	5
<b>3</b>	<b>Produkteinsatz</b>	<b>6</b>
3.1	Anwendungsbereiche . . . . .	6
3.2	Zielgruppen . . . . .	6
3.3	Betriebsbedingungen . . . . .	6
<b>4</b>	<b>Produktumgebung</b>	<b>6</b>
4.1	Software . . . . .	6
4.2	Hardware . . . . .	6
4.3	Orgware . . . . .	6
4.4	Schnittstellen . . . . .	7
<b>5</b>	<b>Funktionale Anforderungen</b>	<b>7</b>
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 . . . . .	10
5.3	KI Client . . . . .	11
5.3.1	Vorbereitungen . . . . .	11
5.3.2	Spielzug . . . . .	12
<b>6</b>	<b>Produktdaten</b>	<b>12</b>
6.1	Allgemein . . . . .	12
6.2	Server . . . . .	12
6.3	Klient . . . . .	13
6.4	KI-Klient . . . . .	13
<b>7</b>	<b>Produktleistungen</b>	<b>13</b>
<b>8</b>	<b>Weitere nichtfunktionale Anforderungen</b>	<b>13</b>
<b>9</b>	<b>Qualitätsanforderungen</b>	<b>14</b>

<b>10 Globale Testfälle und Testszenarien</b>	<b>14</b>
<b>11 Systemmodelle</b>	<b>16</b>
11.1 Übersicht . . . . .	16
<b>12 Benutzungsoberfläche</b>	<b>17</b>
12.1 Hauptmenü . . . . .	17
12.2 Warteraum . . . . .	18
12.3 Spielfeld . . . . .	19
<b>13 Spezielle Anforderungen an die Entwicklungsumgebung</b>	<b>19</b>
<b>14 Zeit- und Ressourcenplanung</b>	<b>20</b>
14.1 Projekt-Untergliederung . . . . .	20
14.2 Ressourcen . . . . .	20
<b>15 Ergänzungen</b>	<b>20</b>
<b>16 Glossar</b>	<b>20</b>

# 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 (R)-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, Einkaufsläden, 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, Häuser und Einkaufsläden 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 Würfeln. Die Summe, die sich aus den Augenzahlen der zwei Würfel ergibt, gibt dem Spieler Auskunft darüber, wie viele Schritte der Spieler 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 auch Abgaben bezahlen muss. Jeder Spieler hat während dem Spiel die Möglichkeit, seine Gebäude und Straßen an die anderen Spieler zu verkaufen, sowie diese für eine Ersteigerung (Auktion) frei zu geben. Kann ein Spieler finanziell seinen Anforderungen nicht mehr nachkommen, so scheidet er automatisch aus. Spieler die vorzeitig ein laufendes Spiel beenden, können ihr Gut 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:  $1024 \times 768$
- Beschleunigte 2D-Grafik

Empfohlen:

- 512 MB Arbeitsspeicher
- 100 MB freien Festplattenspeicher
- 1-GHz-Prozessor
- Bildschirmauflösung:  $1280 \times 1024$
- 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

Bei Aufruf der ausführbaren Datei (Das Programm) soll das Spiel gestartet werden und das Hauptmenü angezeigt werden.

Im Hauptmenü können folgende Punkt ausgewählt:

- Ändern der Spieleinstellungen ( $\rightarrow$  /F11020/)
- Erstellen eines Servers ( $\rightarrow$  /F11050/)

##### /F11020/ Ändern der Spieleinstellungen

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

##### /F11030/ Laden der Einstellungen

Die in den Einstellungen ( $\rightarrow$  /F02/) vorhandenen Daten werden vor dem Anzeigen aus der Datei ( $\rightarrow$  /DB01/) geladen.

##### /F11040/ Speichern der Einstellungen

Die im Einstellungsfenster ( $\rightarrow$  /F02/) eingestellten Einstellungen werden in der Datei ( $\rightarrow$  /DB01/) gespeichert.

##### /F11050/ Erstellen eines Servers (Konfiguration)

Es ist möglich einen Server zu starten. Dazu kann man eine Konfiguration laden ( $\rightarrow$  /F11060/), die Einstellungen ändern und die Einstellungen speichern ( $\rightarrow$  /F11070/). Danach kann der Server dann gestartet werden.

##### /F11060/ Laden der Einstellungen

Die im Konfigurationsfenster des Servers ( $\rightarrow$  /F05/) angezeigten Einstellungen werden vor dem Anzeigen aus der Datei ( $\rightarrow$  /DB02/) geladen.

**/F11070/ Speichern der Einstellungen**

Die in den Einstellungsfenster ( $\rightarrow$  /F05/) eingestellten Einstellungen werden in ( $\rightarrow$  /DB02/) gespeichert.

**/F11080/ Erstellen eines Servers**

Nachdem die Einstellungen durchgeführt wurden ( $\rightarrow$  /F11050/) kann ein Server gestartet werden, dadurch wird ( $\rightarrow$  /F09/) aufgerufen

**/F11090/ Starten eines Servers**

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

**/F11100/ Spiel beitreten**

Im Hauptmenü ( $\rightarrow$  /F01/) kann einem Server beigetreten werden, je nach Benutzerwahl können dann „Direkte Verbindung“ ( $\rightarrow$  /F11110/) oder das „Serverliste“ ( $\rightarrow$  /F11120/) ausgewählt werden. Danach wird ein eventuell vorhandenes Passwort abgefragt, die Namen und Anzahl der Spieler angegeben die sich verbinden wollen und dann die Verbindung aufgebaut ( $\rightarrow$  /F12010/).

**/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 ( $\rightarrow$  /F11130/).

In dieser Liste kann ein Server ausgewählt werden und 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 gibt 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.

### 5.1.2 Bei hergestellter Verbindung

**/F12010/ Initiieren eines Spiels**

Der Warteraum wird angezeigt. Währenddessen kann man sich mit den anderen Spielern unterhalten ( $\rightarrow$  /F17/). Nachdem mindestens zwei Spieler verbunden sind und alle Spieler bereit sind startet das Spiel ( $\rightarrow$  /F18/).

**/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 ein Sonderfeld wird entsprechend dem Sonderfeld vorgegangen. Danach kann der aktive Spieler weitere Aktionen durchführen ( $\rightarrow$  /F120Y0/).

**/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 ( $\rightarrow$  /F120Z0/).

**/F12061/ Straße kaufen**

Da der Spieler nur die Straße kaufen kann, auf der 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/ Aufgaben**

Jeder Spieler kann, sobald er dran 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 und 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 übrig gebliebene kommt an die Bank. Eventuelle 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 ein Chatraum zur Verfügung. Jeder Spieler landet zuerst hier und kann sich auf „bereit“ setzen. Nachdem mindestens zwei Spieler verbunden sind 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 Spieler 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 fängt 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.

#### **/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 dran kommen ( $\rightarrow$  /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 als 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.

## **5.3 KI Client**

### **5.3.1 Vorbereitungen**

#### **/F31001/ Starten des KI-Client**

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

#### **/F31002/ Betreten eines Servers**

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

#### **/F31003/ Laden der KI-Parameter**

Der Klient 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, dass 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 Klient 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

#### /PD02/ Spielregeln (optional)

- automatisch Würfeln?
- automatisch Miete einziehen?
- Bei „frei Parken“ Geld aufnehmen?
- Begrenzung des Spielzugs in Sekunden
- Mindestdauer des Spielzugs in Sekunden

### 6.2 Server

#### /PD03/ Servereinstellungen

- Servername
- Passwort (optional)
- Mindestspielerzahl
- Höchstspieleranzahl

#### /PD04/ Spielstand

- Liste aller teilnehmenden Spieler
- Kontostand jedes Spielers
- Besitz an Spielfeldern und Gebäuden jedes Spielers
- Hypotheken auf Spielfeldern

## 6.3 Klient

/PD05/ Klienten-Einstellungen

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

## 6.4 KI-Klient

/PD06/ 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 "Linkdead": < 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 Java Virtual Machine auf jeder (bedeutenden) Plattform (Mac, Windows, Linux) laufen
- Das Produkt sollte/kann auf JVMs auf Smartphones (zumindest Android) laufen

## 9 Qualitätsanforderungen

- Das Produkt muss zuverlässig Laufen und darf nicht ohne ausführliche Fehlermeldung crashen
- Es müssen durchschnittlich mindestens 5 Spiele in Folge gespielt werden können, ohne dass ein Fehler auftritt
- Probleme mit Netzwerkverbindungen usw müssen Spielintern aufgefangen werden und dürfen nicht zu einer Beendigung des Spieles führen
- 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:

- Aufrufen des Hauptmenüs, Ändern der Spieleinstellungen, Speichern der Spieleinstellungen, Laden der Spieleinstellungen
- Erstellen einer Serverkonfiguration, Speichern der Serverkonfiguration, Laden der Serverkonfiguration, Erstellen eines Servers
- Starten eines Servers, Beitreten (Varianten: über Serverliste (inkl. /F12/, /F1X0Y0/), über IP-Eingabe), Spiel starten, Spiel beenden
- Initiieren eines Spiels, Chatten, Starten des Spiels, Spielen unter Verwendung aller Spielzüge, Aufgeben (Variante: zu Ende spielen).

Folgende Datenkonsistenzen müssen eingehalten werden:

- Zu jedem Zeitpunkt muss ein Spielernamen 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 überprüft 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)
- Druck auf passive Buttons
- 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

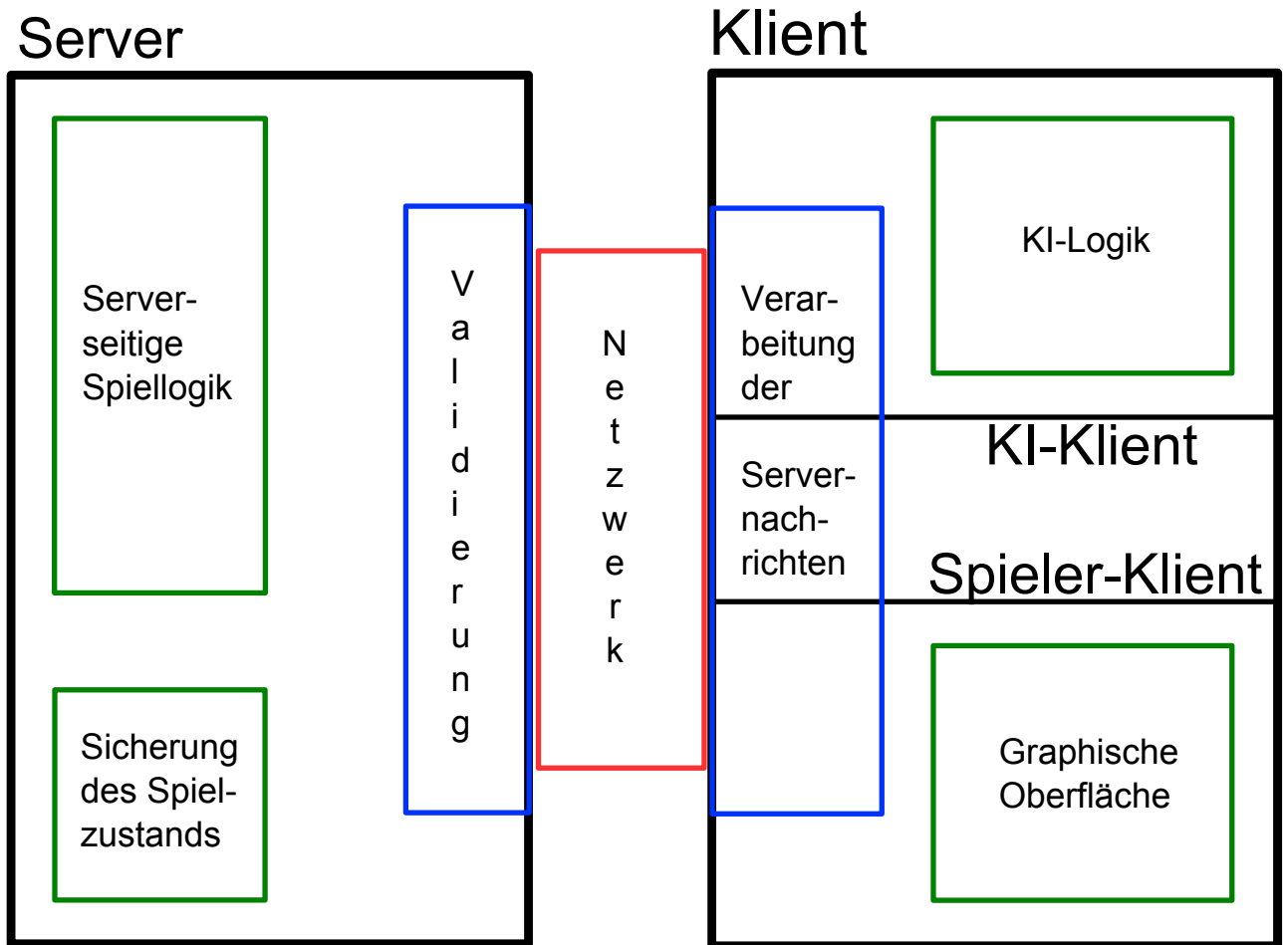
- Korrekte Darstellung auf verschiedenen Betriebssystemen (Windows, Linux, OS X)
- Verschiedene Bildschirmauflösungen
- nicht-standardmäßige Systemschriftgröße

Testszenarien:

- “Erste Inbetriebnahme mit Starten eines Servers”  
Starten des Programms → im Hauptmenü “Spieleinstellungen ändern” aufrufen → Daten eingeben → “Spieleinstellungen speichern” anklicken → “Erstellen eines Servers” 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 → “Beitreten eines Servers” anklicken → “Anzeigen der Serverliste” auswählen → Server auswählen → verbinden → Bereitschaft setzen → Würfeln, Spielzüge durchführen → aufgeben
- “Voller Server”  
Spieler versucht auf vollen Server zu verbinden → erhält Meldung, dass die maximale Teilnehmeranzahl erreicht ist → Möglichkeit, als Zuschauer beizutreten
- “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

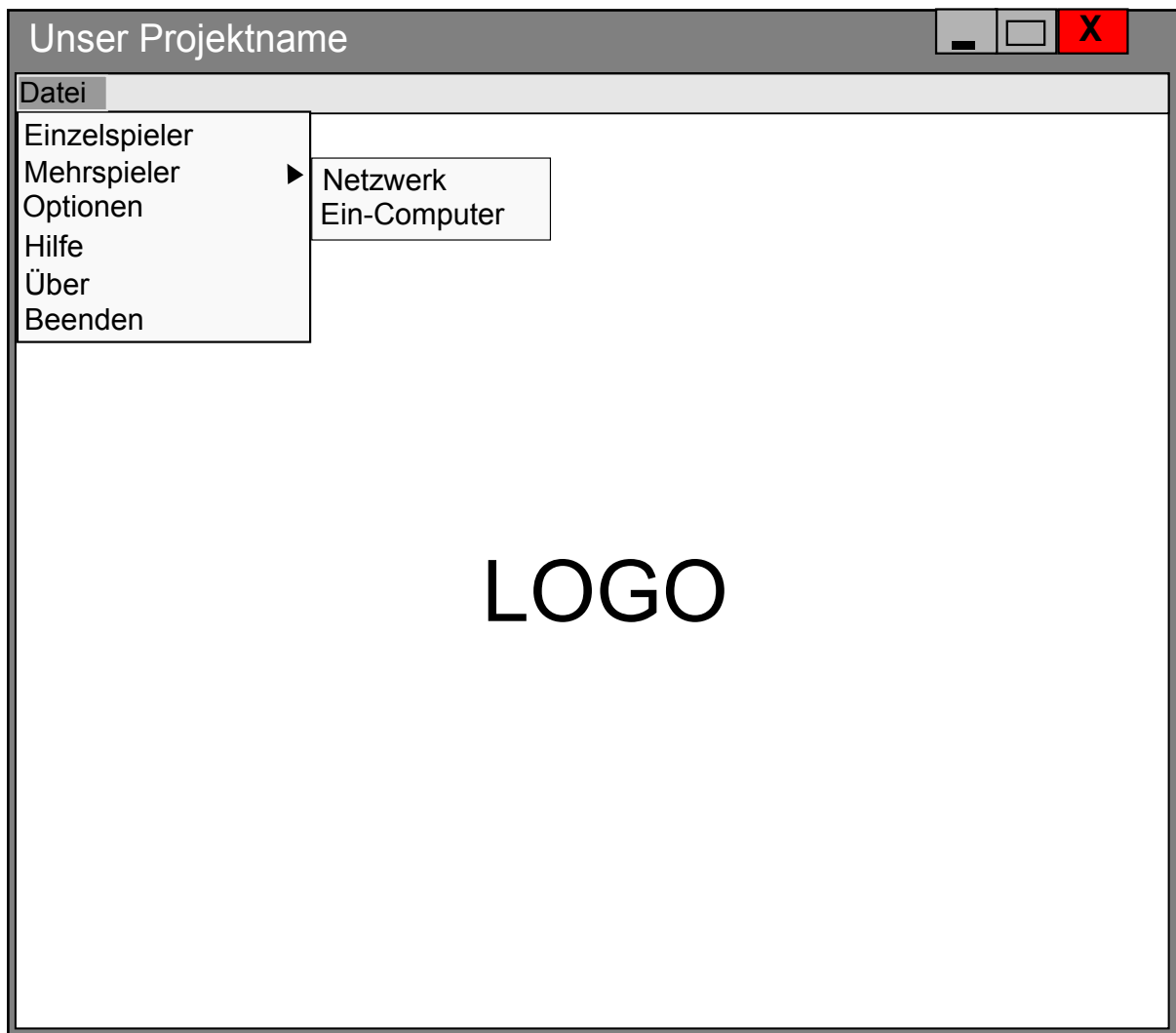
### 11.1 Übersicht





## 12 Benutzungsoberfläche

### 12.1 Hauptmenü



## 12.2 Warteraum

Datei

● xZise 10000 €

○ Vikath 10000 €

● Max 10000 €

○ Jeremias10000 €

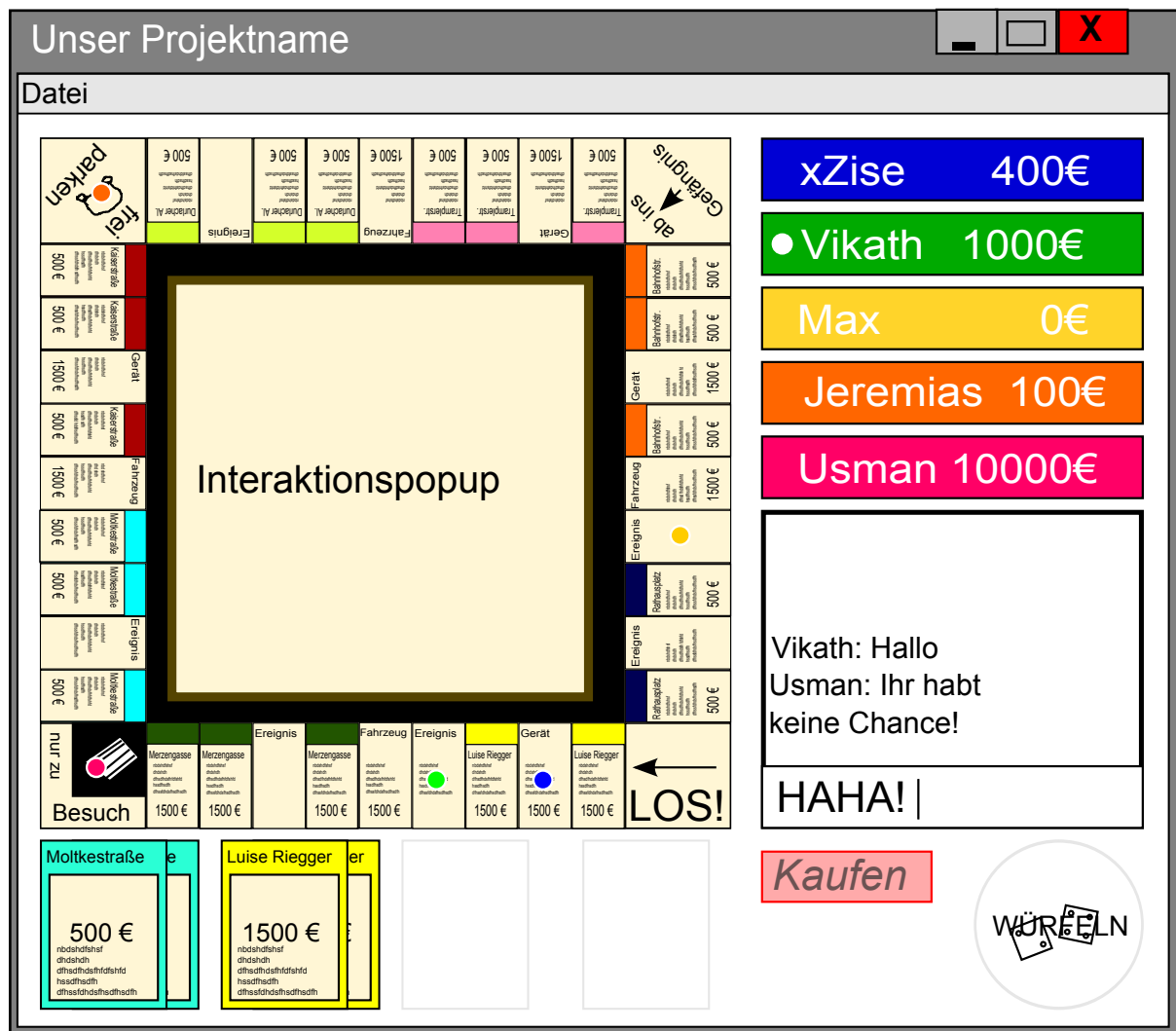
○ Usman

☐ Bereit?

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

Wo???

## 12.3 Spielfeld



## 13 Spezielle Anforderungen an die Entwicklungsumgebung

- Allgemein
  - L<sup>A</sup>T<sub>E</sub>X
  - Versionskontrolle (SVN, Git)
- Entwicklung
  - IDE: Eclipse, Netbeans
- Entwurf
  - Geeignete UML-Tools wie Rational Architect, DIA
- Validierung
  - JUnit
- Teamkommunikation
  - Forum
  - Email(Verteiler)

## 14 Zeit- und Ressourcenplanung

### 14.1 Projekt-Untergliederung

Zu planende und implementierende Module	Zeit	Personen
KI-Spiellogik	75h	NA
Klient-Grundgerüst	40h	NA
GUI - Design+Funktionalität	30h	NA
Menüführung+Funktionalität Klient	20h	NA
serverseitige Spiellogik	100h	NA
Sicherung des Spielzustands	20h	NA
Verifizieren der Klient-Eingabe	10h	NA
Netzwerkanbindung Klient	10h	NA
Netzwerkanbindung Server	10h	NA

### 14.2 Ressourcen

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

## 15 Ergänzungen

## 16 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