

Dokumentation Catan-Protokoll

Protokoll Version 1.0

1 Änderungen gegenüber Version 0.3

- Bei Handel annehmen Feldname korrigiert: **"Mitspieler"** statt **"Spieler"**
- Erweiterbare Spielwelt (für 5–6 Spieler) durch Koordinatensystem
- Spieler um Felder für größte Rittermacht und längste Handelsstraße erweitert.

2 Grundlagen

Die Daten werden über einen persistenten TCP-Stream übertragen. Der Port ist dabei nicht festgelegt. Das Protokoll verwendet einen Strom von JSON-Objekten, als Zeichensatz wird dabei UTF-8 verwendet. Objekte *müssen* dabei so kodiert werden, dass sie selbst keine Zeichenwechsel enthalten (optionale Zeilenwechsel müssen weggelassen werden, in Zeichenketten verwenden Sie **"\n"**). Ein Zeilenwechsel darf ausschließlich als Trennsymbol für Objekte verwendet werden (gestalten Sie aber ihren Parser hierfür möglichst tolerant, damit er auch mit fehlerhaften Nachrichten umgehen kann). Nach jedem Objekt *müssen* Sie einen Zeilenwechsel schicken.

Unterschiedliche Nachrichtentypen sind durch sogenannte Wrapper-Objekte modelliert. Ein Objekt darf dabei stets nur eine Nachricht enthalten.

Die allgemeine Form einer Nachricht sieht folgendermaßen aus:

```
{ "Nachrichtentyp" : {  
  "Attribut"       : "Wert",  
  "2. Attribut"   : "Wert\nZweite Zeile"  
} }
```

Statt einem primitiven **"Wert"** können hier jedoch auch komplexere JSON-Objekte auftreten. Zeilenwechsel müssen bei der Serialisierung weggelassen oder kodiert werden, so dass die Nachricht dann beispielsweise wie folgt verschickt wird:

```
{"Nachrichtentyp":{"Attribut":"Wert","2. Attribut":"Wert\nZweite Zeile"}}
```

3 Kompatibilität

Ihre Implementierung braucht nur die neueste Protokollversion unterstützen.

Ihr Client soll mit *jedem* Server funktionieren, der dieses Protokoll korrekt implementiert. Ebenso soll ihr Server (zumindest im Kompatibilitätsmodus) mit *jedem* Client spielbar sein. Insbesondere gilt dies für die KI-Spieler.

Eigene Erweiterungen können Sie gerne vornehmen, allerdings muss ihr Client und Server über einen Kompatibilitätsmodus verfügen, der Interoperabilität gewährleistet.

4 Hinweise zu JSON

- Beachten Sie, dass Strings in dem Protokoll in der Regel *sensitiv* sind bezüglich der Groß- und Kleinschreibung. Das bedeutet, "Spiel starten" und "Spiel Starten" sind nicht gleich. Implementierungen *können* solche Fehler tolerieren, aber um Interoperabilität zwischen Ihren Spielen zu erreichen müssen Sie sich an die hier vorgegebene Schreibweise halten.
Nutzen Sie den Testserver, und *loggen* Sie empfangene sowie versendete Nachrichten, um ihre Kompatibilität zu prüfen!
- Nachrichten *sollen* als eine einzige Zeile verschickt werden! Formatierte Ausgaben wie in dieser Dokumentation *soll* ihr Programm möglichst auch verstehen können, aber nicht versenden.
- Verlassen Sie sich nicht darauf, dass ein einzelner `read()` Aufruf die vollständige Nachricht liefert. Das Betriebssystem kann die Nachrichten fragmentiert übertragen. Füllen Sie einen Puffer so lange, bis die empfangene Nachricht ein vollständiges JSON-Objekt ergibt (ein korrekter Server sowie Client *sollte* anschließend auch einen Zeilenwechsel senden).
- Die Verwendung von Bibliotheken wie JSON.org, Jackson, GSON, etc. hierfür ist sinnvoll. Keinesfalls sollte das Protokoll manuell per `print` erzeugt werden. Ein automatisches Mapping von Objekten nach JSON kann dabei aber schwierig werden, da es von vielen Nachrichten zwei Arten geben muss (verdeckte Karten!)
- Beachten Sie, dass Daten *fragmentiert* übertragen werden können. Dann liefert ein `read` *nicht* die vollständige Nachricht! Manche der oben genannten Bibliotheken verfügen über eine Streaming-API, die sehr gut für unser Protokoll geeignet ist und direkt über den Socket (bspw. via `Channels.newReader`, `Channels.newWriter`) kommunizieren kann.
- Beachten Sie die UTF-8 Kodierung von Umlauten. Empfehlenswert ist es, Ihr gesamtes Projekt in Eclipse als UTF-8 zu konfigurieren, um Probleme mit Windows zu vermeiden (OSX und Linux verwenden normalerweise bereits UTF-8).

5 Verbindungsaufbau

Nach dem Aufbau der TCP-Verbindung sendet der Server an den Client seine Versions- und Protokollinformationen.

```
{ "Hallo" : { "Version" : "...", "Protokoll" : "1.0" } }
```

Anschließend antwortet der Client mit einer entsprechenden Nachricht.

```
{ "Hallo" : { "Version" : "..."} }
```

Sollte der Client die Protokollversion des Servers nicht unterstützen, so ist die Verbindung abubrechen. Für das Praktikum *müssen* Sie Ihren Gruppennamen in der Versionsnummer verwenden, wie im folgenden Beispiel:

```
{ "Hallo" : { "Version" : "JavaFXClient 1.0 (RuhendeRebellionen)" } }
```

Wenn Ihr Client computergesteuert ist, kennzeichnen Sie ihn auch mittels "(KI)".

Anschließend erhält der Client vom Server eine eindeutige Spielernummer zugewiesen:

```
{ "Willkommen" : { "id" : 42 } }
```

Als Spielernummern dürfen nur positive 31-bit Integer-Werte verwendet werden.

Anschließend verschickt der Server für den neu verbundenen Client eine Statusmeldung (entsprechend Abschnitt 9.1 und Abschnitt 6.5) an alle Spieler. Name und Farbe können dabei noch nicht oder auf `null` gesetzt sein, so dass folgende Statusnachricht zulässig ist:

```
{"Statusupdate":{"Spieler":{"id":42,"Status":"Spiel starten"}}
```

Der neu verbundene Client bekommt zusätzlich für alle anderen Spieler eine Statusmeldung (entsprechend Abschnitt 9.1 und Abschnitt 6.5), so dass er anzeigen kann welche Namen und Farben bereits gewählt sind, und wer das Spiel bereits gestartet hat.

6 Objekte

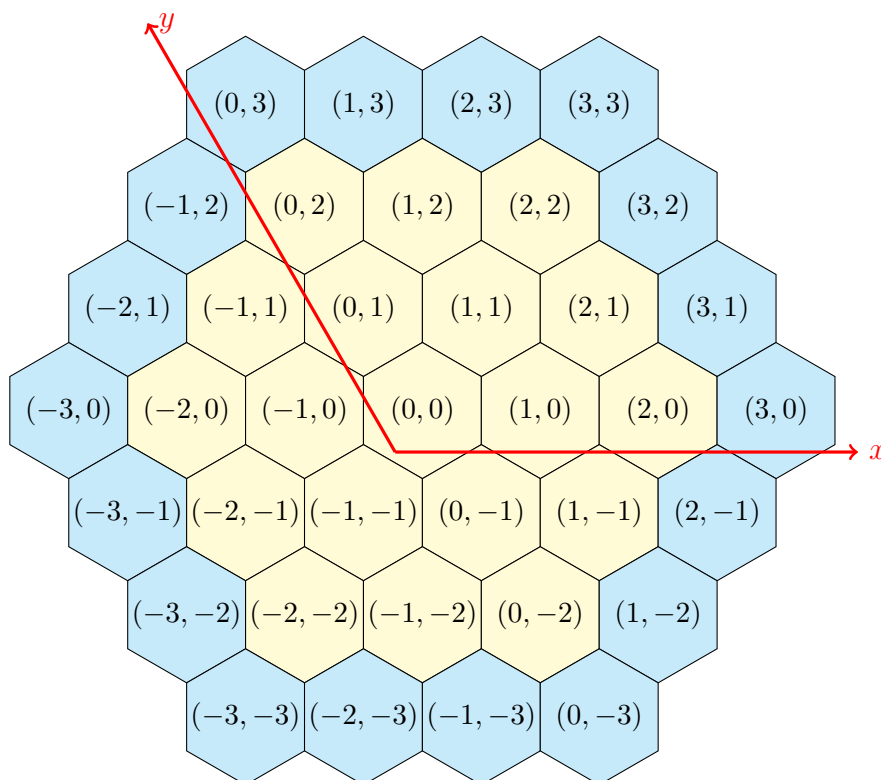
Nachrichten können verschiedene Objekte enthalten. Diese werden als JSON-Objekte übertragen.

6.1 Felder

Um erweiterbare Spielwelten zu ermöglichen, wird statt der festen Feldnamen jetzt ein 2-dimensionales Koordinatensystem verwendet. Dabei zeigt die x -Achse wie gewohnt nach rechts, die y -Achse ist jedoch leicht schief, um die hexagonale Struktur abzubilden.

Der Mittelpunkt der Insel ist dabei stets $(0, 0)$ oder zwischen $(0, 0)$ und $(1, 0)$.

Für die 5–6 Spieler Erweiterung (optional) wird eine größere Spielwelt benötigt.



Felder werden dann repräsentiert als ein Objekt:

```
{
  "Ort" : { "x" : -2, "y" : 2 },
  "Typ" : "Ackerland",
  "Zahl" : 2
}
```

Wobei das Attribut **"Ort"** die Koordinaten in obigem Koordinatensystem angibt und **"Typ"** die Werte **"Ackerland"**, **"Huegelland"**, **"Weideland"**, **"Wald"**, **"Gebirge"**, **"Wueste"** und **"Meer"**

annehmen kann. "Zahl" ist der Wert des Zahlenchips der auf einem Feld liegt, und welcher bestimmt, bei welchem Wurf das Feld Rohstoffe erwirtschaftet.

6.2 Gebäude

Gebäude sind modelliert durch einen "Eigentuemer" (Nummer des Spielers, dem das Gebäude gehört), einem "Typ" (mögliche Werte: "Strasse", "Dorf" und "Stadt") und einem "Ort" an dem sich das Gebäude befindet. Orte sind modelliert durch die Buchstaben der angrenzenden Felder (eine Straße liegt immer genau zwischen zwei Feldern, Orte und Städte liegen zwischen drei Feldern).

```
{
  "Eigentuemer" : 42,
  "Typ"         : "Strasse",
  "Ort"         : [
    { "x" : -2, "y" : 0 },
    { "x" : -1, "y" : 0 }
  ]
}
```

6.3 Häfen

Häfen haben einen "Ort" der genau wie der von Straßen definiert ist. Der "Typ" eines Hafens kann die folgenden Werte annehmen: "Holz Hafen", "Lehm Hafen", "Wolle Hafen", "Erz Hafen", "Getreide Hafen" und "Hafen" (für 3:1 Handel).

```
{
  "Ort" : [
    { "x" : 1, "y" : 2 },
    { "x" : 0, "y" : 2 }
  ],
  "Typ" : "Hafen"
}
```

6.4 Karte

Die Karte besteht aus Feldern, einer Liste von Gebäuden (anfangs oft leer), Häfen sowie der aktuellen Position des Räubers:

```
{
  "Felder" : [ ], // Array von Feldern
  "Gebaende" : [ ], // Array von Gebaeuden
  "Haefen" : [ ], // Array von Haefen
  "Raeuber" : { // Feldposition des Raeubers
    "x" : 0, "y" : 0
  }
}
```

6.5 Spieler

Der Zustand eines Spielers:

```
{
  "id"           : 42,
  "Farbe"        : "Orange",
  "Name"         : "Princess Leia Organa",
  "Status"       : "Spiel starten",
  "Siegpunkte"   : 0,
  "Rohstoffe"    : { }, // Rohstoffe-Objekt
  "Rittermacht"  : 0,    // Nur ausgespielte Ritter-Karten
  "Entwicklungskarten" : { }, // Entwicklungskarten-Objekt
  "Groesste Rittermacht" : false,
  "Laengste Handelsstrasse" : false
}
```

Jeder Spieler soll nur die eigenen Rohstoffe (als **Rohstoffe** Objekt) mitgeteilt bekommen, von anderen Spielern soll er nur die Gesamtzahl als **"Unbekannt"** vom Server mitgeteilt bekommen. **"Rittermacht"** gibt dabei die Anzahl der ausgespielten Ritterkarten wieder (ausschlaggebend für die größte Rittermacht). Ausgespielte Siegpunkte werden direkt in die Siegpunkte eingerechnet. Verdeckt gehaltene Karten sollen anderen Spielern nur als Gesamtzahl ausgegeben werden.

"Status" gibt Aufschluss darüber, welche Aktion der Server von einem Spieler als nächstes erwartet. Folgende Statusmeldungen sind in Protokoll Version 1.0 spezifiziert:

"Spiel starten"	Der Spieler hat das Spiel noch nicht gestartet (Farbe und Namen ändern, Spiel starten)
"Wartet auf Spielbeginn"	Der Spieler wartet bis alle das Spiel gestartet haben (Abwarten und Tee trinken)
"Dorf bauen"	Der Spieler darf ein (kostenloses) Dorf bauen (Siehe: "Bauen" , mit "Typ" : "Dorf")
"Strasse bauen"	Der Spieler darf eine (kostenlose) Straße bauen (Siehe: "Bauen" , mit "Typ" : "Strasse")
"Wuerfeln"	Der Spieler muss als nächstes würfeln (Siehe: "Wuerfeln")
"Karten wegen Raeuber abgeben"	Der Spieler muss Karten abgeben (Siehe: "Karten abgeben")
"Raeuber versetzen"	Der Spieler muss den Räuber versetzen (Siehe: "Raeuber versetzen")
"Handeln oder Bauen"	Der Spieler ist am Zug (Siehe Abschnitt 10)
"Bauen"	Außerordentliche Bauphase (nur 5–6 Spieler)
"Warten"	Der Spieler muss auf andere Spieler warten
"Verbindung verloren"	Die Verbindung zum Spieler wurde getrennt. (Der Server wird das Spiel beenden.)

Das Feld **"Siegpunkte"** soll die bekannten Siegpunkte eines Spielers enthalten. Dies beinhaltet Siegpunkte durch Dörfer, Städte, größte Rittermacht, längste Handelsstraße und eigene

Siegpunktkarten (fremde Siegpunktkarten sollen jedoch nicht enthalten sein).

6.6 Rohstoffe

Rohstoffe-Objekte beschreiben über welche Rohstoffe ein Spieler verfügt, aber auch welche Rohstoffe er erhält oder tauschen möchte.

```
{ "Holz" : 0, "Lehm" : 0, "Wolle" : 0, "Getreide" : 0, "Erz" : 0 }
```

Rohstoffe, die nicht vorhanden sind, dürfen dabei weggelassen werden.

Von fremden Spielern sollen die Rohstoffe verdeckt bleiben, und dürfen nur als Gesamtzahl als "Unbekannt" übermittelt werden.

```
{ "Unbekannt" : 0 }
```

Dies betrifft auch Aktionen wie das Stehlen eines Rohstoffs durch den Räuber.

6.7 Entwicklungskarten

Entwicklungskarten-Objekte spiegeln die dem Spieler zur Verfügung stehenden Karten wieder:

```
{  
  "Ritter"      : 0,  
  "Strassenbau" : 0,  
  "Monopol"     : 0,  
  "Erfindung"  : 0,  
  "Siegpunkt"  : 0  
}
```

Auch hier dürfen nicht vorhandene Karten weggelassen werden, und von fremden Spielern darf nur die Gesamtzahl als "Unbekannt" übermittelt werden.

```
{ "Unbekannt" : 0 }
```


7 Allgemeine Nachrichten

7.1 Bestätigungen und Fehler

Der Server muss jede Aktion des Clients bestätigen. Wenn eine Aktion des Spielers nicht erfolgreich durchgeführt werden konnte, so muss ein Fehler verschickt werden (Sie werden sich aber nicht darauf verlassen können, dass jeder Server die gleichen Meldungen verschickt). Hat eine Aktion weitere Nachrichten zur Folge, so sollen *erst* die anderen Nachrichten verschickt werden, abschließend dann eine "OK"-Nachricht:

```
{ "Serverantwort" : "OK" }  
{ "Serverantwort" : "... " }
```

7.2 Chat senden

Um eine Chat-Nachricht an die Mitspieler zu senden verwenden Sie folgende Nachricht:

```
{ "Chatnachricht senden" : {  
  "Nachricht" : "Help me, Obi-Wan Kenobi."  
} }
```

7.3 Chat empfangen

Schickt ein Client eine Chatnachricht, so wird diese vom Server verteilt.

```
{ "Chatnachricht" : {  
  "Absender" : 42,  
  "Nachricht" : "Help me, Obi-Wan Kenobi."  
} }
```

Verschickt der Server eine Chatnachricht, so fehlt das "Absender" Feld.

8 Konfiguration und Spielstart

Bevor ein Spiel gestartet werden kann, muss jeder Spieler einen Namen und eine Farbe wählen. Ein Spiel kann nur gestartet werden, wenn sich 3 oder 4 Spieler (mit Erweiterungen später auch 5 oder 6 Spieler) verbunden haben, und jeder eine andere Farbe gewählt hat.

Zum Setzen des Namens und zum Wählen der Farbe dient der Nachrichtentyp **"Spieler"**:

```
{ "Spieler" : {  
  "Name"      : "Princess Leia Organa",  
  "Farbe"     : "Orange"  
} }
```

Zulässige Farben sind dabei **"Rot"**, **"Orange"**, **"Blau"**, **"Weiss"** sowie für die 5-6 Spieler Erweiterung (falls unterstützt) zusätzlich **"Gruen"** und **"Braun"**.

Bei jeder Änderung verschickt der Server ein Statusupdate entsprechend Abschnitt 9.1.

8.1 Spielbereitschaft

Jeder Spieler muss dem Server mitteilen, dass das Spiel beginnen kann. Dazu muss *jeder* Spieler die folgende (leere) Nachricht an den Server senden:

```
{ "Spiel starten" : { } }
```

Der Server soll diese Nachricht nur dann mit **"OK"** akzeptieren, wenn die Farbe noch nicht vergeben ist. Andernfalls soll ein Fehler gemeldet werden:

```
{ "Fehler" : {  
  "Meldung" : "Farbe bereits vergeben"  
} }
```

8.2 Spielstart

Haben alle verbundenen Spieler (3 oder 4) das Spiel gestartet, verschickt der Server eine Nachricht mit der Karte (siehe Abschnitt 6.4) an alle Spieler:

```
{ "Spiel gestartet" : {  
  "Karte"           : // Objekt vom Typ Karte  
} }
```

Anschließend wird der Server die Reihenfolge der Spieler festlegen (in Protokoll Version 1.0 würfelt der Server automatisch die Reihenfolge aus. Das Alter der Spieler wird nicht beachtet, da die KIs kein Alter haben. Siehe auch: <http://www.catan.de/faqs/9>) und die initiale Bauphase einleiten.

8.3 Spielende

Hat ein Spieler das Spiel gewonnen, so verschickt der Server eine Nachricht:

```
{ "Spiel beendet" : {  
  "Nachricht"      : "Spieler Princess Leia Organa hat das Spiel gewonnen.",  
} }
```

```
"Sieger"      : 42  
} }
```

9 Nachrichten des Servers im Spiel

9.1 Statusupdate eines Spielers

Wenn sich der Zustand eines Spielers ändert (beispielsweise er am Zug ist), so sendet der Server ein Statusupdate. Diese Nachricht enthält das aktualisierte Spieler-Objekt (siehe Abschnitt 6.5). In der initialen Bauphase wird der Server dem Spieler beispielsweise den Status für denjenigen Spieler auf **"Dorf bauen"** und **"Strasse bauen"** setzen, der am Zug ist.

```
{ "Statusupdate" : {  
  "Spieler"      : // Spieler-Objekt  
} }
```

Diese Nachricht soll vom Server dazu genutzt werden, den allgemeinen Spielablauf (wer ist am Zug etc.) zu steuern, insbesondere für KI-Spieler. Gültige Statusmeldungen hierfür sind im Abschnitt 6.5 spezifiziert. Insbesondere ist ein Status wie bspw. **"Wuerfeln"** als Aufforderung zu sehen, den nächsten Zug zu machen.

9.2 Würfeln

Hat ein Client gewürfelt, so sendet der Server das Ergebnis.

```
{ "Wuerfelwurf" : {  
  "Spieler"      : 42,  
  "Wurf"         : [ 3, 4 ]  
} }
```

9.3 Ertrag

Bekommt ein Spieler beispielsweise durch Würfeln neue Rohstoffe, so erhält er vom Server eine **"Ertrag"** Nachricht, die ein **Rohstoffe**-Objekt (Abschnitt 6.6) enthält.

```
{ "Ertrag"       : {  
  "Spieler"      : 42,  
  "Rohstoffe"    : // Rohstoffe-Objekt  
} }
```

Durch Würfeln erhaltene Rohstoffe werden vom Server allen Spielern bekanntgegeben.

9.4 Kosten

Verbraucht (bspw. durch Bauen) oder verliert (bspw. durch den Räuber) ein Spieler Rohstoffe, so erhält er vom Server eine **"Kosten"** Nachricht, die ein **Rohstoffe**-Objekt (Abschnitt 6.6) enthält.

```
{ "Kosten"       : {  
  "Spieler"      : 42,  
  "Rohstoffe"    : // Rohstoffe-Objekt  
} }
```

Baukosten werden vom Server allen Spielern bekanntgegeben.

9.5 Räuber versetzt

Wurde der Räuber versetzt, so wird ein Objekt verschickt, das neben dem Standort auch den stehlenden- und den bestohlenen Spieler enthalten kann:

```
{ "Raeuber versetzt" : {  
  "Spieler"          : 42,  
  "Ort"              : { "x" : 0, "y" : 1 },  
  "Ziel"             : 13  
} }
```

9.6 Bauvorgang

Hat ein Spieler ein Gebäude gebaut, so sendet der Server eine Nachricht an alle Spieler, mit dem Objekt des neuen Gebäudes.

```
{ "Bauvorgang" : {  
  "Gebaende"       : // Objekt vom Typ Gebaende  
} }
```

Ein bestehendes Dorf an der gleichen Stelle wird dabei ggf. durch eine Stadt ersetzt.

9.7 Entwicklungskarten

Kauft ein Spieler eine Entwicklungskarte, so bekommt er diese per Nachricht zugewiesen (es wird nicht zwischen Ritterkarten und Entwicklungskarten unterschieden).

```
{ "Entwicklungskarte gekauft" : {  
  "Spieler"          : 42,  
  "Entwicklungskarte" : "Ritter"  
} }
```

Das Feld `"Entwicklungskarte"` muss dabei für andere Spieler als `Unbekannt` versteckt werden.

9.8 Entwicklungskarten ausspielen

Wenn ein Spieler eine Entwicklungskarte ausspielt, so verteilt der Server die in Abschnitt 12 spezifizierten Nachrichten.

9.9 Binnenhandel

Die vom Server beim Binnenhandel verschickten Nachrichten sind ausführlich im separaten Abschnitt 11 dokumentiert.

9.10 Rittermacht und Handelsstraße

Erlangt ein Spieler die größte Rittermacht, oder die längste Handelsstraße, so verschickt der Server eine Nachricht

```
{ "Laengste Handelsstrasse" : {  
  "Spieler" : 42  
} }
```

beziehungsweise

```
{ "Groesste Rittermacht" : {  
  "Spieler" : 42  
} }
```

Baut ein Spieler eine gleich lange Handelsstraße, so behält stets der erste Spieler die Handelsstraße! Ebenso bei der Rittermacht – nur eine echt größere Rittermacht führt zum Wechsel.

Verliert ein Spieler die längste Handelsstraße, *ohne* dass ein anderer Spieler sie bekommt (Sonderfall, falls die einzige Straße der Länge 5 unterbrochen wird, siehe Almanach), so versendet der Server die gleiche Nachricht, aber mit einem leeren Objekt:

```
{ "Laengste Handelsstrasse" : { } }
```

Beachten Sie bei der Berechnung der längsten Handelsstraße die Sonderfälle, die in den Spielregeln diskutiert werden. (Siehe auch: <http://www.catan.de/faqs/9>)

10 Nachrichten des Clients im Spiel

10.1 Würfeln

Um zu Würfeln, senden Sie folgende (leere) Nachricht:

```
{ "Wuerfeln": { } }
```

10.2 Räuber: Rohstoffe abgeben

Hat ein Spieler eine 7 gewürfelt, so muss *jeder* Spieler der mehr als 7 Karten hat, die Hälfte davon abgeben. Dazu sendet der Server ein Statusupdate mit "**Karten wegen Räuber abgeben**" an die betroffenen Spieler; diese senden anschließend eine Nachricht an den Server mit den abzugebenden Rohstoffen.

```
{ "Karten abgeben" : {  
  "Abgeben"       : // Rohstoffe-Objekt  
} }
```

Für jeden Spieler verschickt der Server eine "**Kosten**"-Nachricht, die jedoch für andere Spieler nur die Anzahl als "**Unbekannt**", nicht die genau abgegebenen Rohstoffe enthält.

10.3 Räuber versetzen

Ein Spieler, der eine 7 gewürfelt hat, muss den Räuber versetzen (Status: "**Räuber versetzen**"). Dazu schickt er eine Nachricht an den Server mit der gewünschten neuen Position, sowie (optional) der Nummer des Spielers, der bestohlen werden soll. (Das Feld "**Ziel**" soll ausgelassen werden, falls kein anderer Spieler ein angrenzendes Gebäude hat)

```
{ "Raeuber versetzen" : {  
  "Ort"           : { "x" : 0, "y" : 1 },  
  "Ziel"          : 13  
} }
```

Hat der Spieler dabei einen Rohstoff erbeutet, so verschickt der Server passende "**Ertrag**" (Abschnitt 9.3) und "**Kosten**" (Abschnitt 9.4) Nachrichten. Nur die beiden betroffenen Spieler sehen, um welchen Rohstoff es sich dabei handelt. Die anderen Spieler werden nur darüber informiert, dass 1 Rohstoff gestohlen wurde.

10.4 Zug beenden

Um den Spielzug zu beenden:

```
{ "Zug beenden" : { } }
```


10.5 Bauen

Um eine Straße, ein Dorf oder eine Stadt zu bauen senden Sie folgende Nachricht:

```
{ "Bauen" : {  
  "Typ" : "Dorf",  
  "Ort" : [  
    { "x" : -2, "y" : 2 },  
    { "x" : -1, "y" : 2 },  
    { "x" : -1, "y" : 1 }  
  ]  
} }
```

Sofern es sich nicht um kostenlose Bauvorgänge handelt, verschickt der Server anschließend eine **"Kosten"**-Nachricht (Abschnitt 9.4). Hierbei dürfen die Kosten allen Spielern mitgeteilt werden.

10.6 Entwicklungskarte kaufen

Um eine Entwicklungskarte zu kaufen:

```
{ "Entwicklungskarte kaufen" : { } }
```

Der Server verschickt anschließend eine **"Kosten"**-Nachricht (Abschnitt 9.4), sowie die gekaufte Entwicklungskarte (Abschnitt 9.7). Hierbei dürfen die Kosten allen Spielern mitgeteilt werden.

10.7 Entwicklungskarten ausspielen

Entwicklungskarten sind detailliert im Abschnitt 12 beschrieben.

10.8 Seehandel

Um Rohstoffe über die Häfen (je nach Hafen zur Rate von 4:1, 3:1 oder 2:1) zu handeln, schickt der Spieler eine Nachricht an den Server:

```
{ "Seehandel" : {  
  "Angebot" : // Rohstoffe-Objekt,  
  "Nachfrage" : // Rohstoffe-Objekt  
} }
```

Auch hier verschickt der Server passende **"Ertrag"** (Abschnitt 9.3) und **"Kosten"** (Abschnitt 9.4) Nachrichten, die die Rohstoffveränderung durchführen. Beim Handeln sollen die gehandelten Rohstoffe öffentlich sichtbar sein.

10.9 Binnenhandel

Binnenhandel ist ein komplexer Vorgang, der detailliert im Abschnitt 11 beschrieben ist.

11 Binnenhandel

11.1 Handel anbieten

Um einen Handel anzubieten schickt der Spieler ein Handelsangebot an den Server:

```
{ "Handel anbieten" : {  
  "Angebot"      : // Rohstoffe-Objekt,  
  "Nachfrage"   : // Rohstoffe-Objekt  
} }
```

Das Handelsangebot wird vom Server mit einer eindeutigen **"Handel id"** versehen, und an die Mitspieler verteilt. Diese können sich nun überlegen, ob sie auf den Handel eingehen würden.

```
{ "Handelsangebot" : {  
  "Spieler"      : 42,  
  "Handel id"   : 0,  
  "Angebot"     : // Rohstoffe-Objekt,  
  "Nachfrage"   : // Rohstoffe-Objekt  
} }
```

Ein Gegenangebot der Mitspieler ist im Protokoll nicht vorgesehen, aber Sie können den Chat für Verhandlungen nutzen!

11.2 Handelsangebot annehmen

Ist ein Spieler bereit, das Handelsangebot anzunehmen, signalisiert er die Bereitschaft mit der Bestätigung an den Server:

```
{ "Handel annehmen" : {  
  "Handel id"   : 0,  
  "Annehmen"   : true  
} }
```

Das Feld **"Annehmen"** kann auf **false** gesetzt werden, um einen Handel explizit abzulehnen.

Auch diese Aktion wird vom Server an alle Spieler verteilt (es wird offen gehandelt, und es können mehrere Bereitschaft signalisieren):

```
{ "Handelsangebot angenommen" : {  
  "Mitspieler" : 42,  
  "Handel id"  : 0,  
  "Annehmen"   : true  
} }
```

Aus diesen Angeboten kann nun der aktive Spieler eines auswählen.

11.3 Handel durchführen

Der aktive Spieler kann nun auswählen, welchen Handelsvorgang er abschließen kann (es könnten ja mehrere Spieler Handelsbereitschaft signalisiert haben!):

```
{ "Handel abschliessen" : {  
  "Handel id"           : 0,  
  "Mitspieler"         : 2  
} }
```

Bei erfolgtem Handel verschickt der Server eine Bestätigung an alle Spieler.

```
{ "Handel ausgeführt" : {  
  "Handel id"           : 0,  
  "Spieler"            : 1,  
  "Mitspieler"         : 2  
} }
```

Zudem verschickt der Server passende **"Ertrag"** (Abschnitt 9.3) und **"Kosten"** (Abschnitt 9.4) Nachrichten, um die Bestandsveränderung zu signalisieren. Es wird offen gehandelt.

11.4 Handel abbrechen

Um ein Handelsangebot (beiderseits möglich) abzuberechnen, verschickt der Spieler die Nachricht:

```
{ "Handel abbrechen" : {  
  "Handel id"           : 0  
} }
```

Der Server verteilt dies an alle Spieler:

```
{ "Handelsangebot abgebrochen" : {  
  "Spieler"            : 42,  
  "Handel id"         : 0  
} }
```

Je nach Absender bedeutet dies:

- Das ursprüngliche Angebot wurde zurückgezogen.
- Die Bereitschaft das Angebot anzunehmen wurde zurückgezogen.

12 Entwicklungskarten ausspielen

Beachten Sie: es darf jede Runde nur eine Entwicklungskarte gespielt werden, und diese Karte darf nicht erst in der aktuellen Runde gekauft worden sein, mit Ausnahme von Siegpunktkarten (diese müssen nicht ausgespielt werden, sondern können aufgedeckt werden um das Spiel zu gewinnen).

Je nach Art der Karte benötigen Sie unterschiedliche Nachrichten.

12.1 Ritter

Das Ausspielen eines Ritters ähnelt dem versetzen des Räubers (Abschnitt 10.3):

```
{ "Ritter ausspielen" : {  
  "Ort"           : { "x" : 0, "y" : 1 },  
  "Ziel"          : 13  
} }
```

Diese Nachricht wird vom Server an alle Clients verteilt, wobei der Server zusätzlich das Feld **"Spieler"** hinzufügt, dass die Spielernummer des Spielers enthält, der die Karte ausgespielt hat:

```
{ "Ritter ausspielen" : {  
  "Spieler"       : 42,  
  "Ort"           : { "x" : 0, "y" : 1 },  
  "Ziel"          : 13  
} }
```

Für geraubte Rohstoffe werden passende **"Ertrag"** (Abschnitt 9.3) und **"Kosten"** (Abschnitt 9.4) Nachrichten vom Server verschickt. Nicht beteiligte Spieler können jedoch nicht sehen, um welchen Rohstoff es sich handelt.

12.2 Straßenbau

Die Straßenbau-Karte enthält die Koordinaten von bis zu zwei Straßen:

```
{ "Strassenbaukarte ausspielen" : {  
  "Strasse 1" : [ { "x" : -2, "y" : 0 }, { "x" : -1, "y" : 0 } ],  
  "Strasse 2" : [ { "x" : -2, "y" : 1 }, { "x" : -1, "y" : 0 } ]  
} }
```

(Die zweite Straße kann weggelassen werden, wenn der Spieler keinen Platz hat, um eine zweite Straße zu bauen, oder so das Limit von 15 Straßen überschreiten würde. Hat der Spieler bereits 15 Straßen, so kann die Karte nicht ausgespielt werden.)

Diese Nachricht wird vom Server an alle Clients verteilt, wobei der Server zusätzlich das Feld **"Spieler"** hinzufügt. Anschließend verschickt er bis zu zwei **"Bauvorgang"** Nachrichten (Abschnitt 9.6).

```
{ "Strassenbaukarte ausspielen" : {  
  "Spieler" : 42,  
}
```

```

    "Strasse 1" : [ { "x" : -2, "y" : 0 }, { "x" : -1, "y" : 0 } ],
    "Strasse 2" : [ { "x" : -2, "y" : 1 }, { "x" : -1, "y" : 0 } ]
} }

```

12.3 Monopol

Bei der Monopol-Karte muss nur ein Rohstoffname angegeben werden:

```

{ "Monopol" : {
  "Rohstoff" : "Holz"
} }

```

Der Server verschickt zunächst diese Nachricht an alle Clients (wobei er zusätzlich das Feld **"Spieler"** hinzufügt), dann entsprechende **"Ertrag"** (Abschnitt 9.3) und **"Kosten"** (Abschnitt 9.4) Nachrichten an alle betroffenen Spieler. Da es offensichtlich ist, um welchen Rohstoff es sich handelt, muss der Server hier nicht den Rohstofftyp zensieren.

```

{ "Monopol" : {
  "Spieler" : 42,
  "Rohstoff" : "Holz"
} }

```

12.4 Erfindung

Bei der Erfindung hingegen können zwei unterschiedliche Rohstoffe geholt werden.

```

{ "Erfindung" : {
  "Rohstoffe" : // Rohstoffe-Objekt
} }

```

Diese Nachricht wird vom Server an alle Clients verteilt (wobei er zusätzlich das Feld **"Spieler"** hinzufügt), um den Vorgang nachvollziehbar zu machen. Anschließend verschickt er eine passende **"Ertrag"** (Abschnitt 9.3) Nachricht. Diese darf (und sollte, aus Konsistenzgründen) für fremde Spieler auch nur die Anzahl der Rohstoffe enthalten.

```

{ "Erfindung" : {
  "Spieler" : 42,
  "Rohstoffe" : // Rohstoffe-Objekt
} }

```

12.5 Siegpunkte

Siegpunkte können in dieser Protokollversion nicht ausgespielt werden, sondern werden erst aufgedeckt, wenn man damit das Spiel gewonnen hat.

13 Hinweise

Bei Regelfragen konsultieren Sie bitte die offiziellen Regelauslegungen:

- Spielregeln kompakt: http://www.catan.de/files/downloads/4002051693602_catan_-_das_spiel_0.pdf
- Siedler-Almanach (Ausführliche Regelerläuterung):
http://www.catan.de/files/downloads/Siedler_von_Catan-Almanach.pdf

Folgende Regelentscheidungen haben wir zudem getroffen:

- Es wird ausgewürfelt, welcher Spieler beginnt. (Entsprechend Almanach: “Gründungsphase”)
- Es wird mit dem *variablen* Spielaufbau gespielt (siehe Almanach: “Aufbau, Variabler”), nicht nur mit der Einsteiger-Aufstellung. Der Aufbau des Spielfeldes darf dabei der Holz-Version entsprechen (abwechselnd Wasser und ein Hafen, Häfen müssen vollständig an Land angrenzen), sie müssen nicht die festen Rahmenteile der späteren Ausgaben modellieren.
- Die Trennung zwischen Handels- und Bauphase ist aufgehoben, Häfen dürfen sofort genutzt werden (Entsprechend Almanach: “Handeln und Bauen – Trennung aufgehoben”)

Beachten Sie auch die folgenden Regeldetails:

- Die Würfelzahlen 2–12 treten nicht gleich oft auf. Felder mit einer 6 oder 8 sind wertvoller.
- Der variable Spielaufbau stellt sicher, dass die Zahlen 6 und 8 nie nebeneinander liegen!
- Alle Gebäude, Karten und Rohstoffe sind begrenzt. Falls nicht genug Rohstoffe vorhanden sind, entfällt ggf. der Ertrag. Falls ein Spieler keine Dörfer/Straßen/Städte mehr hat, kann er auch keine mehr bauen.
- Jeder Spieler darf maximal 1 Entwicklungskarte pro Runde ausspielen (Ausnahme: Siegpunkte).
- Gekaufte Entwicklungskarten dürfen erst in der *nächsten* Runde ausgespielt werden.
- Spieler sollen nicht genau wissen, welche Karten (sowohl Rohstoffe als auch Entwicklungskarten) andere Spieler auf der Hand halten: Karten werden verdeckt gehalten. Der Server soll für andere Spieler daher nur jeweils die Gesamtzahlen übermitteln.

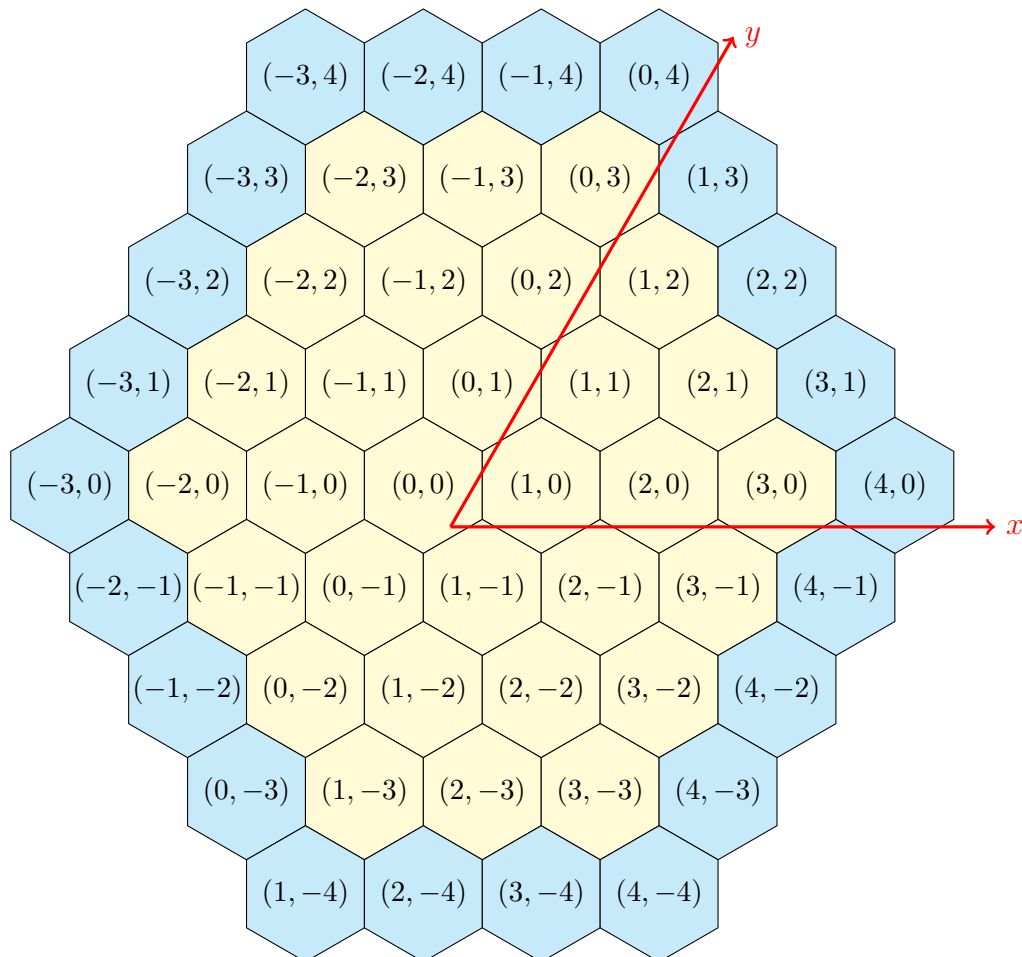
Folgende Hinweise zur Umsetzung in Java:

- Verwenden Sie für JSON eine der vielen Bibliotheken, wie bspw. `org.json`, GSON oder Jackson. Die Verwendung von “Object-Mappern” kann sich dabei aber als schwierig herausstellen, aufgrund der zensierten Rohstoffe u.Ä. (was wir aber nicht lösen wollten indem wir 2x so viele Nachrichten spezifizieren).
- Konfigurieren Sie ihr Java-Projekt mit dem Zeichensatz UTF-8 (Projekteinstellungen). So vermeiden Sie Zeichensatzprobleme mit unterschiedlichen Betriebssystemen (Mac und Linux verwenden i.d.R. UTF-8, Windows die Windows-1252-Kodierung)
- Halten Sie sich an Java-Konventionen, insbesondere was Methoden-, Klassen-, Variablen- und Konstantennamen betrifft. Vermeiden Sie dabei Sonderzeichen.
- Verwenden Sie Konstanten für häufige Zeichenketten, insbesondere für das Protokoll.
- Beobachten Sie ihre Systemlast. Wenn ihr Spiel eine moderne CPU zu 100% auslastet,

haben Sie eine fehlerhafte Schleife (Endlosschleife?) irgendwo in Ihrem Programm. Verwenden Sie `jvisualvm` um herauszufinden, wo ihr Programm zu viel Rechenzeit verbraucht.

14 Anhang: 5-6 Spieler-Modus

Falls Sie diesen Modus unterstützen wollen, verwenden Sie folgenden Spielfeldaufbau:



Verfügbare Felder, sowie die Reihenfolge der Zahlenchips zur Vergabe der Würfelzahlen entnehmen Sie bitte den Webseiten:

- <http://www.siedeln.de/faq/pdf/356.pdf> (Material)
- <http://www.siedeln.de/faq/pdf/365.pdf> (Zahlenchips).