

Dokumentation Catan-Protokoll

Protokoll Version 0.4

Erich Schubert und Johannes Lohrer

23.12.2014

1 Änderungen gegenüber Version 0.3

- Das Feld **"Status"** des Objektes **"Spieler"** ist jetzt explizit spezifiziert.
So kann der Client erkennen, welche Aktion der Server von ihm erwartet.
- Das Ausspielen von Entwicklungskarten wird vom Server mitgeteilt.

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 sollen dabei per Zeilenwechsel ("**\n**") voneinander getrennt werden. Zeilenwechsel innerhalb von Nachrichten sollen vermieden oder entsprechend ersetzt werden, um Kompatibilitätsprobleme zu vermeiden.

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"  
} }
```

Statt einem primitiven **"Wert"** können hier jedoch auch komplexere JSON-Objekte auftreten.

3 Kompatibilität

Ihre Implementierung braucht nur die neueste Protokollversion unterstützen.

Ihr Client soll mit *jedem* Server funktionieren, der dieses Protokoll korrekt implementiert. Ebenso soll ich 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 Verbindungsaufbau

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

```
{ "Hallo" : { "Version" : "...", "Protokoll" : "0.4" } }
```

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" : "SwingClient 0.4 (SparsameWitze)" } }
```

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

Anschließend erhält der Client vom Server eine eindeutige Spielernummer zugewiesen:

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

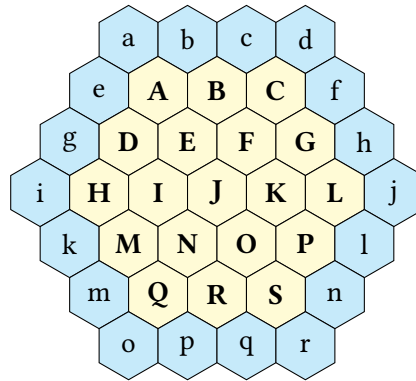
Als Spielernummern dürfen nur positive 31-bit Integer-Werte verwendet werden.

5 Objekte

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

5.1 Felder

Der Kunde hat sich gewünscht das Spielfeld wie folgt zu adressieren:



Felder werden dann repräsentiert als ein Objekt:

```
{  
  "Ort"   : "A",  
  "Typ"   : "Ackerland",  
  "Zahl"  : 2  
}
```

Wobei das Attribut "Ort" die Zelle in obiger Karte angibt und "Typ" die Werte "Ackerland", "Hügelland", "Weideland", "Wald", "Gebirge", "Wüste" 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.

5.2 Gebäude

Gebäude sind modelliert durch einen "Eigentümer" (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).

```
{  
  "Eigentümer" : 42,  
  "Typ"        : "Straße",  
  "Ort"        : "HI"  
}
```

5.3 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" : "fG",  
  "Typ" : "Hafen"  
}
```

5.4 Karte

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

```
{  
  "Felder" : [ ], // Array von Feldern  
  "Gebäude" : [ ], // Array von Gebäuden  
  "Häfen" : [ ], // Array von Häfen  
  "Räuber" : "J" // Feldposition des Räubers  
}
```

5.5 Spieler

Der Zustand eines Spielers:

```
{
  "id"           : 42,
  "Farbe"        : "Orange",
  "Name"         : "Postfrosch",
  "Status"       : "Spiel starten",
  "Siegpunkte"   : 0,
  "Rohstoffe"    : 0, // Oder Rohstoffe-Objekt
  "Ritter"       : 0, // Nur ausgespielte Ritter-Karten
  "Entwicklungskarten" : 0 // Oder Entwicklungskarten-Objekt
}
```

Jeder Spieler soll nur die eigenen Rohstoffe (als **Rohstoffe** Objekt) mitgeteilt bekommen, von anderen Spielern soll er nur die Gesamtzahl vom Server erhalten. **"Ritter"** 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 0.4 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")
"Würfeln"	Der Spieler muss als nächstes würfeln (Siehe: "Würfeln")
"Karten wegen Räuber abgeben"	Der Spieler muss Karten abgeben (Siehe: "Karten abgeben")
"Räuber versetzen"	Der Spieler muss den Räuber versetzen (Siehe: "Räuber versetzen")
"Handeln oder Bauen"	Der Spieler ist am Zug (Siehe Abschnitt 9)
"Warten"	Der Spieler muss auf andere Spieler warten (Abwarten, und Tee trinken)
"Verbindung verloren"	Die Verbindung zum Spieler wurde getrennt. (Der Server wird das Spiel beenden.)

5.6 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.

5.7 Entwicklungskarten

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

```
{  
  "Ritter"      : 0,  
  "Straßenbau" : 0,  
  "Monopol"    : 0,  
  "Erfindung"  : 0,  
  "Siegpunkte" : 0  
}
```

Auch hier dürfen nicht vorhandene Karten weggelassen werden.

6 Allgemeine Nachrichten

6.1 Bestätigungen und Fehler

Der Server soll jede Aktion des Clients bestätigen (umgekehrt ist das derzeit nicht vorgesehen). Wenn eine Aktion des Spielers nicht erfolgreich durchgeführt werden konnte, so soll ein Fehler verschickt werden (Sie werden sich aber nicht darauf verlassen können, dass jeder Server die gleichen Meldungen verschickt).

```
{ "OK" : { } }
```

```
{ "Fehler" : { "Meldung" : "..."} }
```

Obwohl die "OK" Nachricht keine Attribute hat, schicken Sie ein leeres Objekt mit. Gegebenenfalls wird es notwendig sein, hier später zusätzliche Attribute einzufügen.

6.2 Chat senden

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

```
{ "Chatnachricht senden" : {  
  "Nachricht" : "Was ist gelb und hüpfte um den Teich?"  
} }
```

6.3 Chat empfangen

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

```
{ "Chatnachricht" : {  
  "Absender" : 42,  
  "Nachricht" : "Was ist gelb und hüpfte um den Teich?"  
} }
```

7 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"      : "Postfrosch",  
  "Farbe"     : "Orange"  
} }
```

Zulässige Farben sind dabei `"Rot"`, `"Orange"`, `"Blau"` und `"Weiß"`.

Wenn alle Spieler versammelt sind, sollen Sie 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 akzeptieren, wenn die Farbe noch nicht vergeben ist. Andernfalls soll ein Fehler gemeldet werden:

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

Haben alle Spieler das Spiel gestartet, verschickt der Server eine Nachricht mit der Karte (siehe Abschnitt 5.4) an alle Spieler:

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

Anschließend wird der Server die Reihenfolge der Spieler festlegen (in Protokoll Version 0.2 würfelt der Server automatisch die Reihenfolge aus, das Alter der Spieler wird nicht beachtet entsprechend http://www.siedeln.de/faq/205_1158_de.html) und die initiale Bauphase einleiten.

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

```
{ "Spiel beendet" : {  
  "Nachricht"      : "Spieler Postfrosch hat das Spiel gewonnen.",  
  "Sieger"         : 42  
} }
```


8 Nachrichten des Servers im Spiel

8.1 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 5.5). 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. Gültige Statusmeldungen hierfür sind im Abschnitt 5.5 spezifiziert.

8.2 Würfeln

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

```
{ "Würfelwurf" : {  
  "Spieler"    : 42,  
  "Wurf"       : 7  
} }
```

8.3 Ertrag

Bekommt ein Spieler beispielsweise durch Würfeln neue Rohstoffe, so erhält er vom Server eine "Ertrag" Nachricht, die ein Rohstoffe-Objekt (Abschnitt 5.6) enthält.

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

8.4 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 5.6) enthält.

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

8.5 Räuber versetzt

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

```
{ "Räuber versetzt" : {  
  "Spieler"      : 42,  
  "Ort"          : "F",  
  "Ziel"         : 13  
} }
```

8.6 Bauvorgang

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

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

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

8.7 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 nur als Zahl (**1**) dargestellt werden, oder muss weggelassen werden, um nicht zu verraten was für eine Karte der jeweilige Spieler bekommen hat.

8.8 Entwicklungskarten ausspielen

Wenn ein Spieler eine Entwicklungskarte ausspielt, so verteilt der Server die in Abschnitt 11 spezifizierten Nachrichten.

8.9 Binnenhandel

Die vom Server beim Binnenhandel verschickten Nachrichten sind ausführlich im separaten Abschnitt 10 dokumentiert.

8.10 Rittermacht und Handelsstraße

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

```
{ "Längste Handelsstraße" : {  
  "Spieler" : 42  
} }
```

beziehungsweise

```
{ "Größte 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:

```
{ "Längste Handelsstraße" : { } }
```

Beachten Sie bei der Berechnung der längsten Handelsstraße die Sonderfälle, die in den Spielregeln diskutiert werden. (Siehe auch: <http://www.siedeln.de/faq/category190.html>)

9 Nachrichten des Clients im Spiel

9.1 Würfeln

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

```
{ "Würfeln": { } }
```

9.2 Ressourcen 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  
} }
```

9.3 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 kann ausgelassen werden, falls kein anderer Spieler ein angrenzendes Gebäude hat)

```
{ "Räuber versetzen" : {  
  "Ort"           : "F",  
  "Ziel"          : 13  
} }
```

Hat der Spieler dabei einen Rohstoff erbeutet, so verschickt der Server passende "**Ertrag**" (Abschnitt 8.3) und "**Kosten**" (Abschnitt 8.4) Nachrichten.

9.4 Bauen

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

```
{ "Bauen" : {  
  "Typ"      : "Dorf",  
  "Ort"      : "ABE"  
} }
```

Sofern es sich nicht um kostenlose Bauvorgänge handelt, verschickt der Server anschließend eine "**Kosten**"-Nachricht (Abschnitt 8.4).

9.5 Entwicklungskarte kaufen

Um eine Entwicklungskarte zu kaufen:

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

9.6 Entwicklungskarten ausspielen

Entwicklungskarten sind detailliert im Abschnitt 11 beschrieben.

9.7 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 8.3) und "Kosten" (Abschnitt 8.4) Nachrichten, die die Rohstoffveränderung durchführen.

9.8 Binnenhandel

Binnenhandel ist ein komplexer Vorgang, der detailliert im Abschnitt 10 beschrieben ist.

9.9 Zug beenden

Um den Spielzug zu beenden:

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

10 Binnenhandel

10.1 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!

10.2 Handelsangebot annehmen

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

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

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

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

Aus diesen Angeboten kann nun der aktive Spieler eines auswählen.

10.3 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 erfolgreichem Handel verschickt der Server eine Bestätigung an alle Spieler.

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

Zudem verschickt der Server passende "**Ertrag**" (Abschnitt 8.3) und "**Kosten**" (Abschnitt 8.4) Nachrichten, um die Bestandsveränderung zu signalisieren.

10.4 Handel abbrechen

Um ein Handelsangebot (beiderseits möglich) abzubrechen, 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.

11 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.

11.1 Ritter

Das Ausspielen eines Ritters ähnelt dem versetzen des Räubers (Abschnitt 9.3):

```
{ "Ritter ausspielen" : {  
  "Ort"           : "F",  
  "Ziel"          : 13  
} }
```

Diese Nachricht wird vom Server an alle Clients verteilt.

11.2 Straßenbau

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

```
{ "Straßenbaukarte ausspielen" : {  
  "Straße 1"           : "HI",  
  "Straße 2"           : "DI"  
} }
```

(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. Anschließend verschickt er bis zu zwei "Bauvorgang" Nachrichten (Abschnitt 8.6).

11.3 Monopol

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

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

Der Server verschickt zunächst diese Nachricht an alle Clients, dann entsprechende "Ertrag" (Abschnitt 8.3) und "Kosten" (Abschnitt 8.4) Nachrichten an alle betroffenen Spieler. Da es offensichtlich ist, um welchen Rohstoff es sich handelt, muss der Server hier nicht den Rohstofftyp zensieren, darf es aber (es ist also zulässig – und aus Konsistenzgründen sinnvoll – nur die Anzahl der Rohstoffkarten bei anderen Spielern zu übermitteln). Der Server soll das "OK" erst verschicken, wenn die Rohstoffe übertragen sind, um die Entwicklung von KIs einfacher zu gestalten.

11.4 Erfindung

Bei der Erfindung hingegen können zwei unterschiedliche Rohstoffe geholt werden.

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

Diese Nachricht wird vom Server an alle Clients verteilt, um den Vorgang nachvollziehbar zu machen. Anschließend verschickt er eine passende "Ertrag" (Abschnitt 8.3) Nachricht. Diese darf (und sollte, aus Konsistenzgründen) für fremde Spieler auch nur die Anzahl der Rohstoffe enthalten.

11.5 Siegpunkte

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

12 Hinweise

Bei Regelfragen konsultieren Sie bitte die offiziellen Regelauslegungen:

- Spielregeln kompakt: http://www.catan.de/files/downloads/Siedler_von_Catan-Basis-3_4-Spielregel.pdf
- Siedler-Almanach (Ausführliche Regelerläuterung):
http://www.catan.de/files/downloads/Siedler_von_Catan-Almanach.pdf
- Siedeln.de FAQ (A1-A3):
<http://www.siedeln.de/faq/category4.html>

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 (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, damit der erste Spieler keinen zu großen Vorteil hat.
- 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. (Siehe http://www.siedeln.de/faq/159_356_de.html)
- Jeder Spieler darf maximal 1 Entwicklungskarte pro Runde ausspielen (Ausnahme: Siegpunkte).
- Gekaufte Entwicklungskarten dürfen erst in der *nächsten* Runde ausgespielt werden, ebenso darf ein neuer Hafen erst in der *nächsten* Runde zum Handeln genutzt werden (siehe Almanach).
- 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 und ähnlicher Asymmetrien (die 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.