Dokumentation Catan-Protokoll

Protokoll Version 0.2

Erich Schubert Julian Busch

05.12.2016

1 Änderungen gegenüber Version 0.1

- Der Räuber ist jetzt im Protokoll spezifiziert
- Seehandel und Binnenhandel sind jetzt im Protokoll spezifiziert
- Kosten analog zu Ertrag modelliert
- Spielende möglich

2 Einschränkungen

Die folgenden Einschränkungen wurden für Protokoll-Version 0.2 vorgenommen:

- Es sind noch keine Entwicklungskarten verfügbar.
- Es gibt noch keine längste Handelsstraße oder größte Rittermacht.

Diese Einschränkungen werden in der nächsten Versionen des Protokolls aufgehoben.

3 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\nZweite Zeile"
} }
```

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

4 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.

5 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!
- 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 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).

6 Verbindungsaufbau

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

```
{ "Hallo" : { "Version" : "..." , "Protokoll" : "0.2" } }
```

Anschließend antwortet der Client mit einer entsprechenden Nachricht.

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

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

```
{ "Hallo" : { "Version" : "JavaFXClient 0.2 (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 } }
```

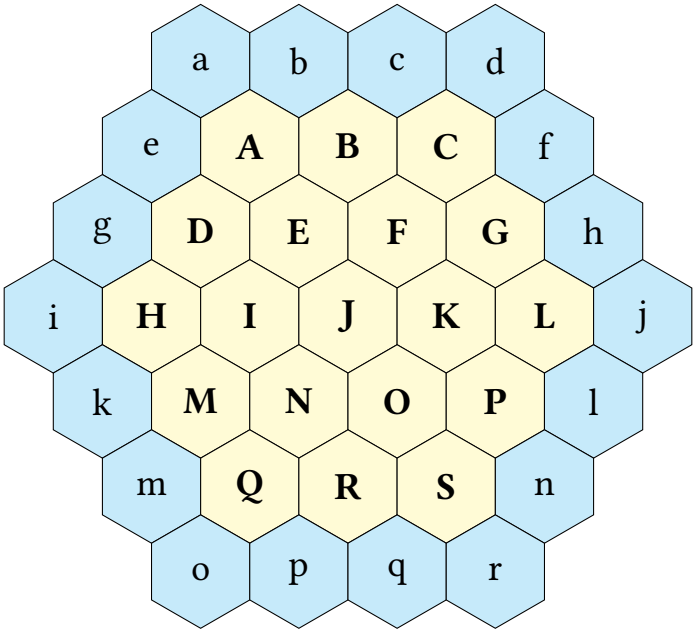
Als Spielernummern dürfen nur positive 31-bit Integer-Werte verwendet werden.

7 Objekte

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

7.1 Felder

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



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.

7.2 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"
}
```

7.3 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"
}
```

7.4 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
}
```

7.5 Spieler

Der Zustand eines Spielers:

```
{
  "id"          : 42,
  "Farbe"       : "Orange",
  "Name"        : "Princess Leia Organa",
  "Status"      : "Spiel starten",
  "Siegpunkte" : 0,
  "Rohstoffe"   : { } // Rohstoffe-Objekt
}
```

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.

`"Status"` gibt Aufschluss darüber, welche Aktion der Server von einem Spieler als nächstes erwartet. Folgende Statusmeldungen sind in Protokoll Version 0.2 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 11)
`"Warten"`	Der Spieler muss auf andere Spieler warten
`"Verbindung verloren"`	Die Verbindung zu einem (anderen) Spieler wurde getrennt. (Der Server wird das Spiel beenden.)

7.6 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 }
```

8 Allgemeine Nachrichten

8.1 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).

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

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

8.2 Chat senden

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

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

8.3 Chat empfangen

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

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

9 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"` 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 7.4) 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, da die KIs kein Alter haben. Siehe auch: `http://www.catan.de/faq/737-spielablauf-muessen-wir-uns-immer-die-regel-halten-dass-der-aelteste-die-erste-siedlung`) und die initiale Bauphase einleiten.

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
} }
```

10 Nachrichten des Servers im Spiel

10.1 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 7.5). 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 7.5 spezifiziert.

10.2 Würfeln

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

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

10.3 Ertrag

Bekommt ein Spieler beispielsweise durch Würfeln neue Rohstoffe, so erhält er vom Server eine `"Ertrag"` Nachricht, die ein `Rohstoffe`-Objekt (Abschnitt 7.6) enthält.

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

10.4 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 7.6) enthält.

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

10.5 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
} }
```

10.6 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.

10.7 Binnenhandel

Die vom Server beim Binnenhandel verschickten Nachrichten sind ausführlich im separaten Abschnitt 12 dokumentiert.

11 Nachrichten des Clients im Spiel

11.1 Würfeln

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

```
{ "Würfeln": { } }
```

11.2 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
} }
```

11.3 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 10.3) und `"Kosten"` (Abschnitt 10.4) Nachrichten.

11.4 Bauen

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

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

11.5 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 10.3) und `"Kosten"` (Abschnitt 10.4) Nachrichten, die die Rohstoffveränderung durchführen.

11.6 Binnenhandel

Binnenhandel ist ein komplexer Vorgang, der detailliert im Abschnitt 12 beschrieben ist.

11.7 Zug beenden

Um den Spielzug zu beenden:

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

12 Binnenhandel

12.1 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!

12.2 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.

12.3 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" : {
  "Spieler"           : 1,
  "Mitspieler"        : 2
} }
```

Zudem verschickt der Server passende `"Ertrag"` (Abschnitt 10.3) und `"Kosten"` (Abschnitt 10.4) Nachrichten, um die Bestandsveränderung zu signalisieren.

12.4 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.