

Netzwerkprotokoll für Settlers of Asgard (Programmierprojekt CS108 Gruppe3) v2.0.0

1. Einleitung

Dieses Dokument beschreibt das Netzwerkprotokoll für das rundenbasierte Strategiespiel Settlers of Asgard. Es basiert auf TCP und verwendet ein textbasiertes, befehlsorientiertes Format mit fester Befehlslänge. Alle Nachrichten bestehen aus Befehlen und Parametern. Zwischen jedem Befehl und Parameter muss ein Dollar-Zeichen (\$) stehen.

2. Nachrichtenformat

Jede Nachricht hat folgendes Format:

<Befehl>\$<Parameter1>\$<Parameter2>\$...

Beispiel:

BILD\$PL01\$Castle\$0001\$0002 Dieser Befehl bedeutet, dass Spieler PL01 eine Struktur vom Typ "Castle" auf das Feld (1,2) bauen möchte.

Feste Befehlslänge: Jeder Befehl hat genau 4 Zeichen. Eine Ausnahme stellen die Nachrichten in der Fehlerbehandlung dar (ERROR, OK). Die Parameter haben eine variable Länge und werden durch Dollar-Symbole getrennt. Falls keine Parameter benötigt werden, bleibt die Nachricht nach dem Dollar-Symbol leer.

Maximale Parameterlängen:

- **Spielername:** maximal 16 Zeichen.
- **Nachricht:** maximal 200 Zeichen.
- Bei Überschreitung dieser Längen wird ein Fehler (Fehlercode 103) zurückgegeben.

3. Befehle

Spielverwaltung

Befehl	Beschreibung	Parameter	Beispiel
JOIN	Spieler tritt bei	Spielername\$SpielID	JOIN\$ThorOdinson\$SP01
EXIT	Spieler verlässt	Spielername\$SpielID	EXIT\$ThorOdinson\$SP01
CHTG	Nachricht senden (global)	SpielerID\$Nachricht	CHTG\$PL01\$HelloWorld!
CHTP	Nachricht senden (privat)	SpielerID1\$SpielerID2\$Nachricht	CHTP\$PL01\$PL02\$HelloWorld!
LIST	Liste der Lobbies		LIST\$
STRT	Spiel starten		STRT\$
STDN	Nachricht an Clients: Server wird beendet		STDN\$
PING	Verbindung prüfen		PING\$
STAT	Spielstatus abrufen	SpielID	STAT\$SP01
SYNC	Synchronisation des Spielzustands anfordern		SYNC\$

Spielmechanik

Befehl	Beschreibung	Parameter	Beispiel
TURN	Startet Zug	SpielerID	TURN\$PL01
ENDT	Beendet Zug	SpielerID	ENDT\$PL01
BUYH	Kauft ein Hexfeld	SpielerID\$X\$Y	BUYH\$PL01\$0001\$0002
BILD	Baut Struktur	SpielerID\$Struktur\$X\$Y	BILD\$PL01\$Castle\$0001\$0002
UPGD	Eine Struktur verbessern	SpielerID\$X\$Y	UPGD\$PL01\$0001\$0002
TRAD	Handeln	Spieler1ID\$Spieler2ID\$RessourcenID1\$RessourcenID2\$Anzahl1\$Anzahl2	TRAD\$PL01\$PL02\$R001\$R002\$0005\$0020
BLNC	Aktuelles Ressourcenkonto abrufen	SpielerID	BLNC\$PL01
RITU	Ritual starten	SpielerID\$RessourcenID\$Anzahl	RITU\$SP01\$R001\$0005
BLES	Segen erhalten	SpielerID\$GottID	BLES\$PL01\$G001

Befehl	Beschreibung	Parameter	Beispiel
CURS	Fluch wirken	<code>Spieler1ID\$Spieler2ID\$GottID</code>	<code>CURS\$PL01\$PL02\$G002</code>
ARTF	Artefakt benutzen	<code>SpielerID\$ArtefaktID</code>	<code>ARTF\$PL01\$AR01`</code>
FIND	Artefakt suchen	<code>SpielerID\$X\$Y</code>	<code>FIND\$PL01\$0001\$0002</code>

4. Fehlerbehandlung

Eine allgemeine Fehlernachricht hat folgendes Format:

`ERROR$<Fehlercode>$<Fehlermeldung>`

Beispiel:

`ERROR100Unbekannter Befehl`

Spezifische Liste der Fehlercodes

Fehlercode	Name	Beschreibung	Beispiel
100	Unbekannter Befehl	Tritt auf, wenn der empfangene Befehl nicht im Protokoll definiert ist.	<code>LEAV\$ThorOdinson\$SP01</code>
101	Falsche Parameteranzahl	Tritt auf, wenn ein Befehl zu wenige oder zu viele Parameter enthält.	<code>JOIN\$ThorOdinson\$SP01\$PL01</code>
102	Ungültiger Parameter	Tritt auf, wenn ein Parameter nicht dem erwarteten Format entspricht.	<code>JOIN\$12345\$SP01</code>
103	Ungültige Länge	Tritt auf, wenn ein Befehl oder Parameter die maximal erlaubte Länge überschreitet.	<code>JOIN\$ThorOdinson123456783456\$SP01;</code>
104	Unzulässige Zeichen	Tritt auf, wenn unzulässige Zeichen (z. B. Semikolons) innerhalb eines Parameters gefunden werden.	<code>JOIN\$Thor\$Odinson\$SP01</code>
105	Unbekannte SpielerID	Tritt auf, wenn eine SpielerID (oder Spielername) nicht existiert oder ungültig ist.	<code>TURN\$PL999</code>
106	Nichtautorisierte Aktion	Tritt auf, wenn ein Spieler versucht, eine Aktion auszuführen, zu der er nicht berechtigt ist.	<code>STRT\$PL02</code>
107	Spiel-ID nicht gefunden	Tritt auf, wenn eine angegebene Spiel-ID nicht existiert oder ungültig ist.	<code>JOIN\$ThorOdinson\$SP999</code>

Protokoll-Antwort bei erfolgreichen Befehlen

Wenn ein Befehl korrekt verarbeitet wurde, schickt der Server eine Bestätigung zurück, damit der Client weiß, dass alles erfolgreich war. Diese Nachricht hat folgendes Format:

`OK$<Befehl>`

Beispiel:

`OK$JOIN`