

# Vorläufiger Schnittstellenentwurf

Autoren:

Kathrin Holzmann	20228
Rainer Hihn	19859
Florian Rosenkranz	19895

## Inhalt

Datenrahmen.....	3
Übersicht über die Nachrichtentypen.....	3
Server.....	4
Login-Thread.....	4
Beschreibung.....	4
Datenrahmen.....	4
Ablaufdiagramm.....	5
PlayerList .....	6
Beschreibung.....	6
Aufbau.....	6
Client-Thread.....	6
Beschreibung.....	6
Datenrahmen.....	6
Ablaufdiagramm.....	8
Loader.....	10
Client.....	11
Command-Thread.....	11
Beschreibung.....	11
Ablaufdiagramm.....	12
Fragenwechsel-Thread.....	12
Datenrahmen.....	12
Ablaufdiagramm.....	13
Listener-Thread.....	14
Beschreibung.....	14
Ablaufdiagramm.....	14
Projektteam.....	15
Zusammensetzung.....	15
Aufgabenverteilung.....	15

## Datenrahmen

```

0          7          15          23
+-----+-----+-----+-----+
| Type      | Length      |
+-----+-----+-----+-----+

```

Type: uint8\_t, gibt die Art der Nachricht an

Length: uint16\_t, Länge der nachfolgenden Zusatzdaten in Bytes

### **Übersicht über die Nachrichtentypen**

Type	Name	Beschreibung	Richtg.
1	LoginRequest	Anmeldung des Clients am Server	C ==> S
2	LoginResponseOK	Anmeldung am Server erfolgreich	C <== S
3	PlayerList	Liste der Spielteilnehmer	C <== S
4	CatalogRequest	Anforderung der Liste der Fragakataloge durch den Client	C ==> S
5	CatalogResponse	Name eines Fragekatalogs (mehrere Nachrichten dieses Typs ergeben die vollständige Liste)	C <== S
6	CatalogChange	Spielleiter hat Katalogauswahl geändert	C <=> S
7	StartGame	Spielleiter möchte Spiel starten	C <=> S
8	QuestionRequest	Anforderung einer Quizfrage durch einen Client	C ==> S
9	Question	Transport einer Quiz-Frage zum Client	C <== S
10	QuestionAnswered	Quiz-Frage wurde beantwortet	C ==> S
11	QuestionResult	Auswertung einer Antwort auf eine Quiz-Frage	C <== S

12	GameOver	Alle Clients sind fertig, Mitteilung über Endplatzierung	C <== S
255	ErrorWarning	Fehlermeldung an den Client	C <== S

Tabelle der Datenrahmentypen und deren Inhalt

## Server

### *Login-Thread*

#### Beschreibung

Dieser Thread wird beim Starten des Servers *geforked*, das heißt automatisch mit gestartet. Sobald eine Anmeldung am Server erfolgreich ist, werden die nötigen Informationen an die Clientliste übertragen. Hierbei wird auch ein separater Client-Thread pro Client gestartet, dieser wartet nun auf Befehle.

#### Datenrahmen

##### LoginResponseOK

0	7	15	23	31
Type	Length	Client-ID		

Type: 2  
 Length: 1  
 Client-ID: uint8\_t, ID des Clients (0 := Spielleiter)

##### PlayerList

Die folgende Nachricht wird aus Platzgründen verkürzt dargestellt. Das Players-Feld ist immer 37\*Spieleranzahl Bytes groß.

0	7	15	23	31
Type	Length	Players .....	=	
=	.....			

Type: 3  
 Length: 37\*Spieleranzahl (maximal 6\*37 = 222)  
 Players: Liste aller derzeit angemeldeten Benutzer  
ErrorWarning

```

0              7              15              23              31
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type      | Length      | Subtype      |
+-----+-----+-----+-----+-----+-----+-----+
| [Message] | ..... |
+-----+-----+-----+-----+-----+-----+-----+

```

Type: 255  
 Length: 1 + Länge(Message)  
 Subtype: uint8\_t  
           0 -> fataler Fehler  
           1 -> Warnung  
 Message: Beschreibung des Fehlers im Textformat (UTF-8), nicht  
           nullterminiert

## Ablaufdiagramm

## Ablaufdiagramm Login-Thread

## PlayerList

### Beschreibung

Speicherung und Verwaltung wichtiger Informationen, die für die Clientkommunikation nötig sind. Während der Spielphase muss der Server diese Liste absteigend nach den Punkteständen der Spieler sortieren.

### Aufbau

32 Bytes	Spielername X (UTF-8, nullterminiert)
4 Bytes	Punktestand Spieler X, vorzeichenlos
1 Byte	ID Spieler X

...

## Client-Thread

### Beschreibung

Der Client-Thread ist die wichtigste Schnittstelle zum Client. Für Jeden der angemeldeten Clients gibt es einen Client-Thread. Die Identifizierung wird über die ClientID zugewiesen

### Datenrahmen

#### CatalogResponse

```

0               7               15               23               31
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type      | Length | [Filename] .. |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type: 5  
 Length: Länge des Dateinamens, oder 0 für Endemarkierung  
 Filename: Dateiname eines Fragekataloges (UTF-8, nicht null-terminiert), oder leer für Endemarkierung

StartGame

```

0           7           15           23           31
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Type      | Length      | [Filename] .. |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Type: 7  
 Length: Länge des Dateinamens  
 Filename: Dateiname des zu spielenden Fragekataloges (UTF-8, nicht nullterminiert), kann beim Versand Server ==> Client (nicht Client ==> Server!) auch leer gelassen werden

Question

Die folgende Nachricht wird aus Platzgründen verkürzt dargestellt. Wenn Fragedaten versendet werden (also nicht die Nachricht verschickt wird, dass es keine Fragen mehr gibt), dann ist das Data-Feld genau 770 Bytes groß.

```

0           7           15           23           31
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Type      | Length      | [Data] ..... =
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
= ..... |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Type: 9  
 Length: 770 oder 0 (falls keine Fragen mehr)  
 Data: Eine Struktur, die wie unten angegeben aufgebaut ist, oder leer (falls keine Fragen mehr)

## Aufbau des Data-Felds:

256 Bytes	Text der Fragestellung (UTF-8, nullterminiert)
128 Bytes	Antworttext 1 (UTF-8, nullterminiert)
128 Bytes	Antworttext 2 (UTF-8, nullterminiert)
128 Bytes	Antworttext 3 (UTF-8, nullterminiert)
128 Bytes	Antworttext 4 (UTF-8, nullterminiert)
2 Bytes	Zeitbegrenzung in Sekunden, vorzeichenlos, in Network Byte Order

QuestionResult

```

0              7              15              23              31
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type          | Length          | Selection          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Correct       |
+---+---+---+---+

```

Type: 11  
Length: 2  
Selection: uint8\_t, Index der zuvor vom Benutzer gewählten Antwort  
(0 <= Selection <= 3), oder 255 falls Zeit für Frage  
abgelaufen  
Correct: uint8\_t, Index der richtigen Antwort (0 <= Correct <= 3)

### GameOver

```

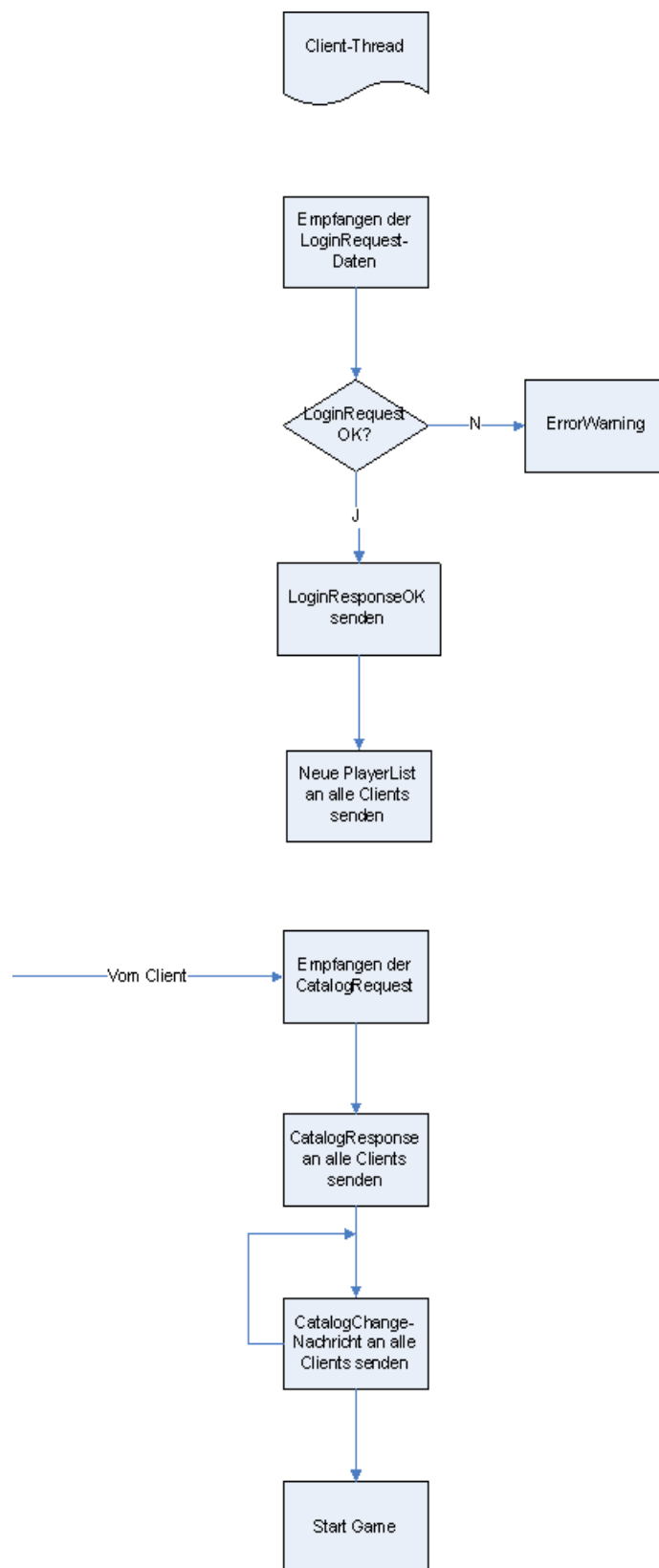
0              7              15              23              31
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type          | Length          | Rank          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

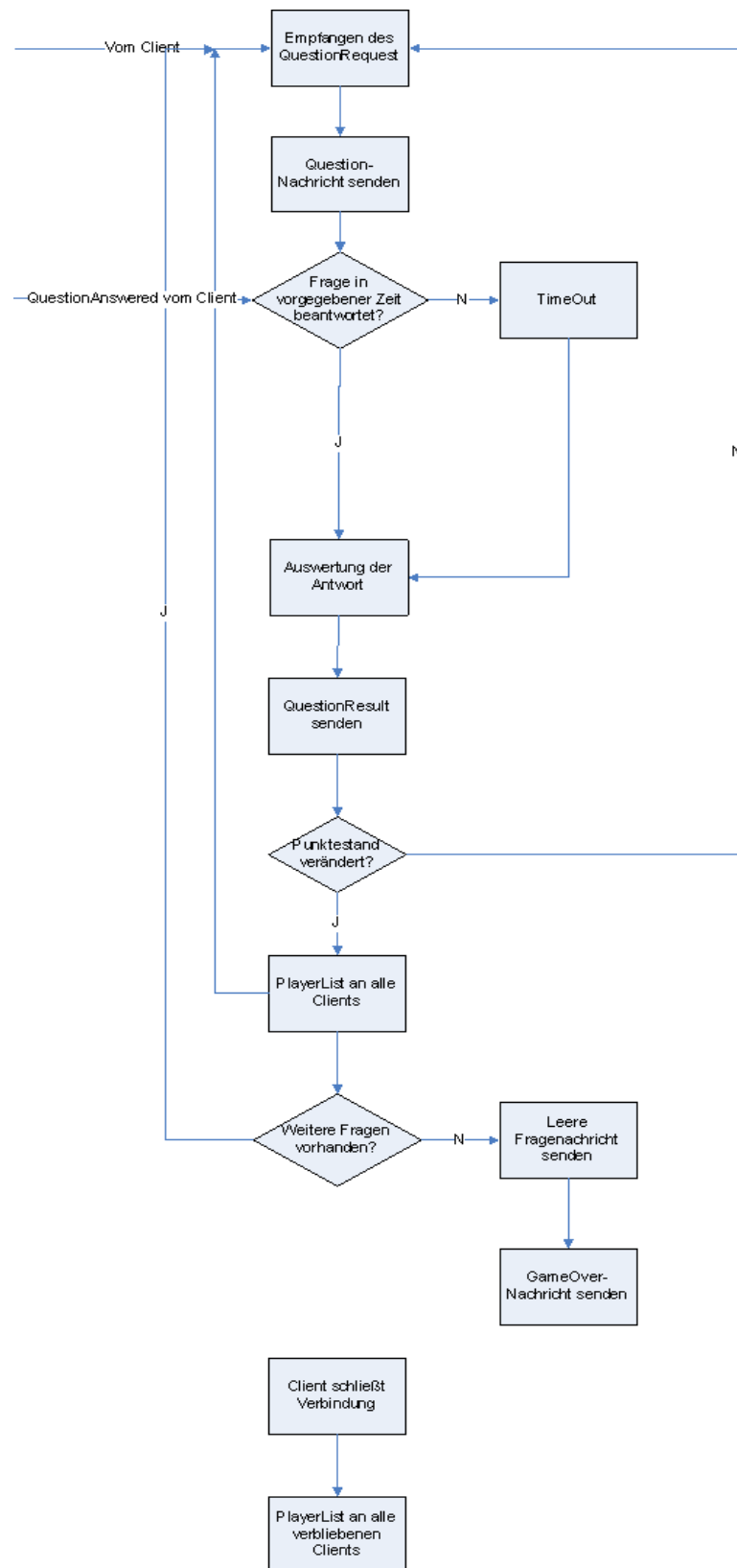
```

Type: 12  
Length: 1  
Rank: uint8\_t, Endposition des Benutzers in der Rangliste  
(1 <= Rank <= 6)

## Ablaufdiagramm







## Loader

Erhält durch den Client-Thread Loader-Kommandos und führt eine Aktion mit anschließender Antwort aus.

Kommando	Bedeutung
LOAD	Fragenkatalog in den Shared Memory Laden
BROWSE	Liste aller Fragenkataloge auf der Standardausgabe ausgeben

Meldung	Bedeutung
LOADED, SIZE = n	Katalog mit n Fragen erfolgreich geladen
ERROR: CANNOT OPEN FILE	Fragenkatalog konnte nicht geöffnet werden
ERROR: CANNOT READ FILE	Fehler beim Lesen aus dem Fragenkatalog
ERROR: INVALID CATALOG	Übergebene Datei ist kein gültiger Fragenkatalog
ERROR: CANNOT USE SHARED MEMORY	Fehler bei der Verwendung des Shared Memory
ERROR: OUT OF MEMORY	Zu wenig freier Hauptspeicher

Übernommen aus "Systemprogrammierung unter Linux, Client- Server Projekt Multiplayer Quiz" Seite 8

# Client

## *Command-Thread*

### Beschreibung

Besitzt die eigentliche Verbindung zum Client-Thread des Servers. Somit besteht eine eindeutige Socketverbindung zwischen Client und Server.

Der Command-Thread sendet die eigentlichen Befehle aus Server- / Client-Thread an den Server.

## Datenrahmen

### LoginRequest

```

0              7              15              23              31
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type          | Length          | Name ..... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: 1  
 Length: Länge des Namens (Length <= 31)  
 Name: Login-Name, UTF-8, nicht nullterminiert, maximal 31 Bytes

### Catalog Request

```

0              7              15              23
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type          | Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: 4  
 Length: 0

### CatalogChange

```

0              7              15              23              31
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type          | Length          | Filename .... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: 6  
 Length: Länge des Dateinamens  
 Filename: Dateiname des gewählten Fragekataloges (UTF-8, nicht nullterminiert)

StartGame

```

0          7          15          23          31
+-----+-----+-----+-----+
| Type      | Length      | [Filename] .. |
+-----+-----+-----+-----+
```

Type: 7  
 Length: Länge des Dateinamens  
 Filename: Dateiname des zu spielenden Fragekataloges (UTF-8, nicht nullterminiert), kann beim Versand Server ==> Client (nicht Client ==> Server!) auch leer gelassen werden

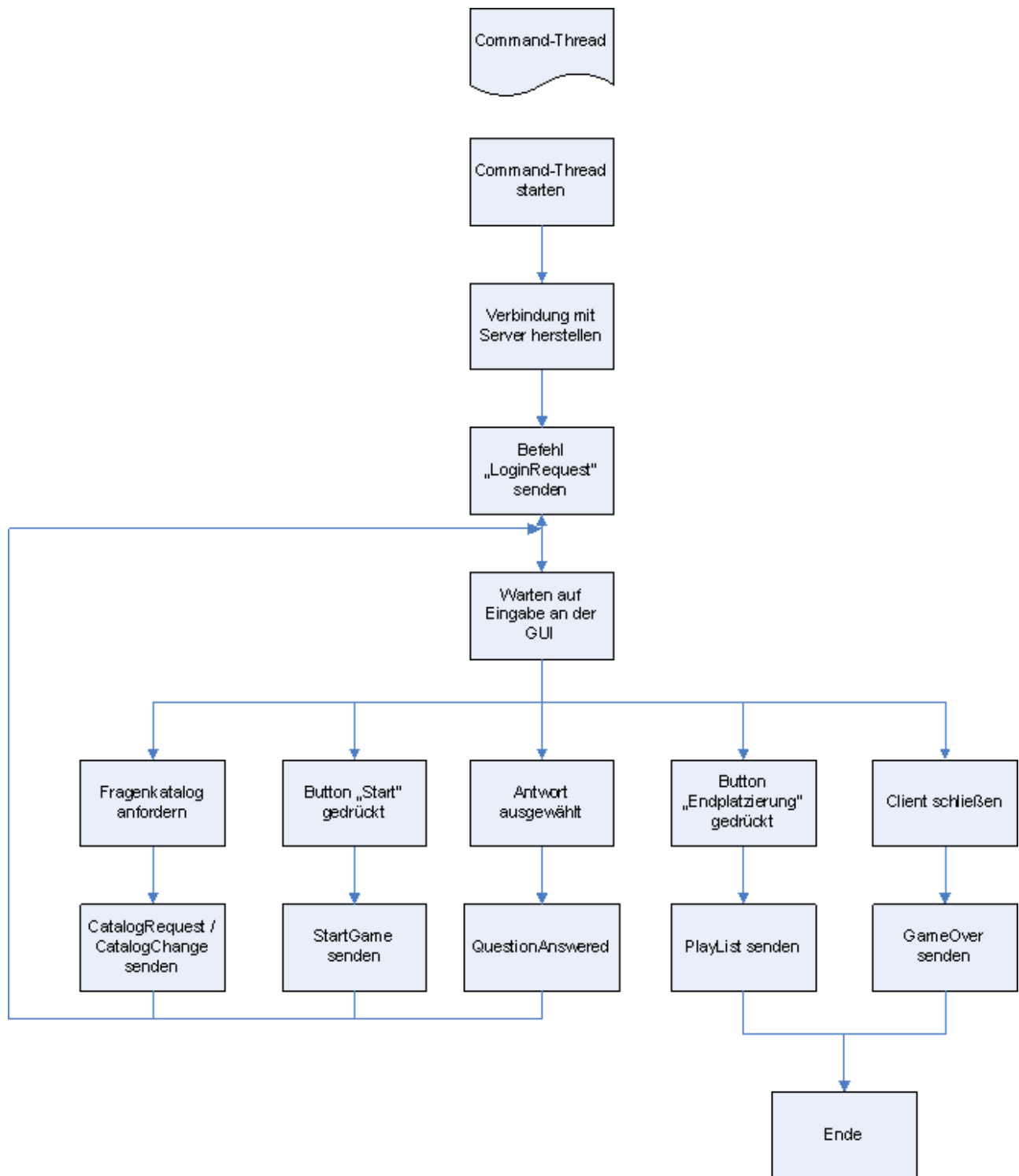
QuestionAnswered

```

0          7          15          23          31
+-----+-----+-----+-----+
| Type      | Length      | Selection      |
+-----+-----+-----+-----+
```

Type: 10  
 Length: 1  
 Selection: uint8\_t, Index der vom Benutzer gewählten Antwortmöglichkeit (0 <= Selection <= 3)

## Ablaufdiagramm



Ablaufdiagramm Command-Thread

## Fragenwechsel-Thread

## Beschreibung

Verantwortlich für die Wartezeit zwischen den Fragen und das Anfordern neuer Fragen.

# Datenrahmen

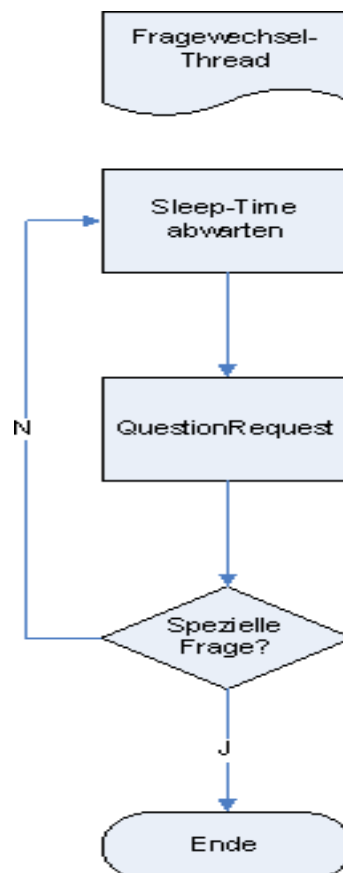
## QuestionRequest

0	7	15	23
+ + + + + + + +	+ + + + + + + +	+ + + + + + + +	+ + + + + + + +
Type	Length		
+ + + + + + + +	+ + + + + + + +	+ + + + + + + +	+ + + + + + + +

```
Type:      8
Length:    0
```

## Ablaufdiagramm

### Ablaufdiagramm Fragenwechsel-Thread

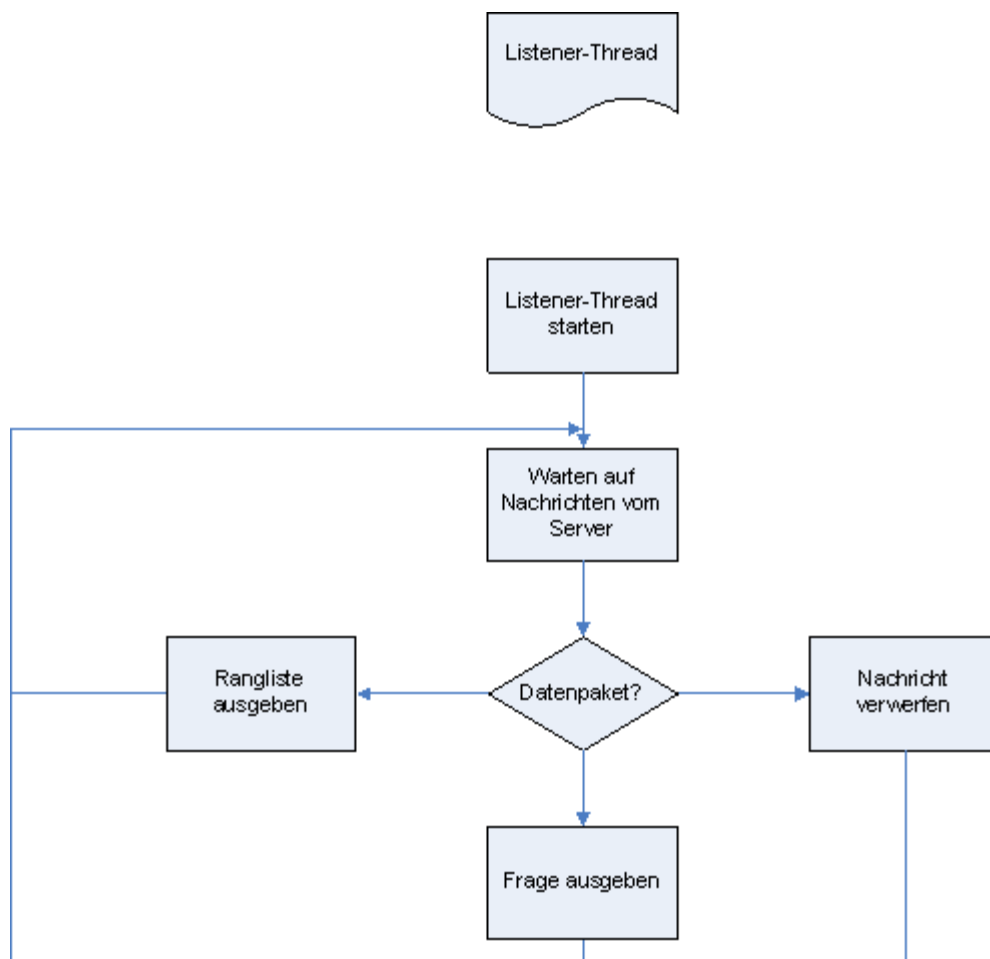


## Listener-Thread

### Beschreibung

Dieser Thread achtet auf Broadcasts vom Server zur Aktualisierung des Punktestands. Außerdem achtet er auf die Eingaben der Benutzer.

### Ablaufdiagramm



Ablaufdiagramm Listener-Thread



## Projektteam

### Zusammensetzung

Teammitglieder: Rainer Hihn, Kathrin Holzmann, Florian Rosenkranz

Teamleiter: Kathrin Holzmann

### Aufgabenverteilung

Rainer Hihn: Login, Vorbereitung, Doku

Florian Rosenkranz: Spiel, Spielende, Logout, Doku

Kathrin Holzmann: Spiel, Spielende, Logout, Doku