Jugend Forscht

## Network

Unser Netzwerk System basiert auf einem Host/Client System. Die Person die die Konferenz startet ist der Host. Jede andere Person die der Konferenz beitritt verbindet sich daher als Client mit dem Host.

Jeder Benutzer der Konferenz bekommt eine User-Id in Form eines bytes zugeordnet. Der Host hat immer die Id 0.

Der Host verfügt über eine Implementation eines TCP sowie eines UDP Servers. Die Clients verbinden sich mit dem Host über TCP und wenn es möglich ist auch über UDP.

## Network Package System

Wir haben ein eigenes Package System entwickelt mit welchem wir die Netzwerk Nachrichten versenden.

Jedes Package hat folgende Signatur:

Header:

* 1 byte (1 bool) der Angibt, ob das Package asynchrone Behandelt werden soll.
* 4 byte (1 Int) der die länge des Inhalts angibt.
* 1 byte der die Packet-Id angibt.
* 1 byte der die User-Id des Erstellers des Packet‘s angibt.

Inhalt:

Nun folgt eine Abfolge von Daten, die für jede Packet-Id unterschiedlich sind. Die Menge dieser folgenden Bytes müssen mit der Inhaltslänge im Header übereinstimmen.

Abstraktion System:

Wir haben eine Abstraktion System zwischen die Implantation von TCP/UDP und dem Rest des Programmes gebaut. So macht es für den Rest kein Unterschied, ob es als Host oder Client läuft.

Es gibt keinen zentralen Server in der Daten-Struktur aller anderen Systeme.

# FileShare:

Um 3D Modelle anzeigen zu können benötigt jeder Nutzer der Konferenz eine Version der Ursprungs Datei. Um diese automatisch zu verteilen haben wir eine System namens FileShare gebaut. Dieses System funktioniert wie folgt.

Jeder Nutzer besitzt eine Liste aller Dateien, die er kennt. Dieser Liste kann er selber eine Datei von seinem Gerät hinzufügen. Die Datei wird dann mit Datei-Pfad, Namen vermerkt. Es wird keine Kopie der Datei angelegt, sodass das hinzufügen fast keine Rechenleistung benötigt. Außerdem ist in jeder mit jeder Datei vermerkt welche Nutzer eine lokale Kopie davon besitzen.

Jeder Nutzer kann nun ein Packet absenden, welches alle anderen Nutzer auffordert ihm eine Liste aller ihrer lokalen Datei-Namen zu senden. Mit diesen Listen vervollständigt der Nutzer seine eigene Liste von Dateien und vermerkt welche Nutzer lokale Kopie davon besitzen.

Laufend wird die Liste der Dateien durchgegangen. Wenn eine Datei eigetragen ist, die der Nutzer nicht lokale vorliegen hat fragt er bei einem der vermerkten Nutzer eine Kopie an. Nun wird ein Daten Austausch zwischen den beiden Nutzern gestartet bei dem die Datei in vielen kleinen Packages zum Nutzer gesendet wird. Wenn die Übertragung erfolgreich war vermerkt der Nutzer auch diese Datei als lokal vorhanden.

In dem FileShare System haben wir versucht ein shared-State System aufzubauen. In diesem können Dateien zugreifbar bleiben, auch wenn der Nutzer die die Datei erstmalig zu Verfügung gestellt hat gar nicht mehr online ist.

## Warum haben gibt es in unserem System keinen zentralen Server sowie es viele andere Konferenz-Tools Programme machen?

Die einfache Antwort ist; er ist nicht notwendig.

Die ausführliche Antwort ist, dass ein zentraler Server zwar viele technische Probleme löst, aber auch einige Probleme mit sich bringt. Einerseits läuft ein Server durchgängig, auch wenn er gerade von niemanden genutzt wird. Wenn man einen Server nutzt, ist man auf eine Internet Verbindung angewiesen.