## Projekt Idee

Die Idee von unserem Projekt hatte Maarten durch seinen Nebenjob bei der Firma Arvico. Die Firma Arvico erstellt unter anderem aus Bauplänen von geplanten Schiffs- und Bauprojekten 3D Modelle und visualisiert diese für Poster und Werbevideos. TODO: Mit Sven absprechen

Dabei war immer eine die offene Frage, wie man die 3D Modelle einfach einem potenzellen Kunden vorstellt. Bei der Vorstellung bieten sich VR-Brillen an, da man mit einer VR-Brille viel besseres Gefühl für Größen bekommt.

Wegen der Corona-Pandemie kann man sich oft nicht persönlich treffen.

Deswegen haben wir uns entschieden ein Konferenzprogramm zu entwickeln, in dem man 3D-Modelle online Vorstellen kann. Dabei loggen sich alle Nutzer in eine virtuelle Umgebung ein in der man in FirstPerson oder VR seinen Charakter bewegen kann.

Es gibt ein 3D-Voicechat System indem man Nutzer nur hört, wenn sie der Nähe sind. So kann man sich leicht in Kleingruppen aufteilen.

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.

Hier führ haben wir die C# nativen TCPListener und UDPListener libaries verwendet. Diese erlauben es einen bytes zu einem anderen Nutzer zu schicken.

## Network Package System

Teil unseres Projekt ist es ein Network kommunik System zu entwickeln, welches die rudimenträren Fuktionen der C# TCP und UDP libaries in ein einfacher nutzbares System verpackt.

Dafür haben ein Package System entwickelt mit welchem wir Netzwerk Packages versenden können. Jedes Package ist mit einer Id als byte difiniert, welche Art von Package es ist. Jede verschiedene Netzwerknachricht die wir senden hat eine eigene Indivuelle Id. Z.B hat so ein neue Postion eines Benutzers in der virtuellen Welt die Id 20 und die Nachricht das ein Benutzer die Konferenz verlassen hat die Id 4.

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.

Das Package System bietet eine Abstraktion ebene die dafür sorgt, das es für die anderen Bereiche egal ist ob der Benutzer eine Host oder Client ist.

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 und 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.