

Technische Dokumentation VR-
Visualization
Virtual Reality, Master Informatik

Matthias Haselmaier, Andreas M. Brunnet

Architektur der Anwendung

Zur Visualisierung der Daten stellt die VR-Anwendung vier Komponenten zur Verfügung, welche hierarchisch aufgebaut sind. Die *Visualizer*-Komponente stellt die Schnittstelle zwischen der UI und der internen Datenstruktur dar. Sie ist dafür zuständig, die Daten aus einer CSV-Datei einzulesen, der UI die nötigen Informationen bereitzustellen und die vom Benutzer gewählte *ScatterplotMatrix* zu erstellen. Die *ScatterplotMatrix*-Komponente wird als Kind des *Visualizers* erstellt. Sie ist für das Erzeugen und Positionieren der einzelnen *Scatterplots* verantwortlich. Die *Scatterplot*-Komponente erstellt ein 3D-Diagramm, in dem die drei übergebenen Dimensionen für alle *DataPoints* gegeneinander aufgetragen werden. Die *DataPoint*-Komponente stellt einen Datenpunkt als Kugel im *Scatterplot* dar. Wie die Szene nach dem Importieren einer CSV-Datei aussieht, ist auf [Abb. 1](#) gezeigt. Im Folgenden werden die Komponenten genauer beschrieben.

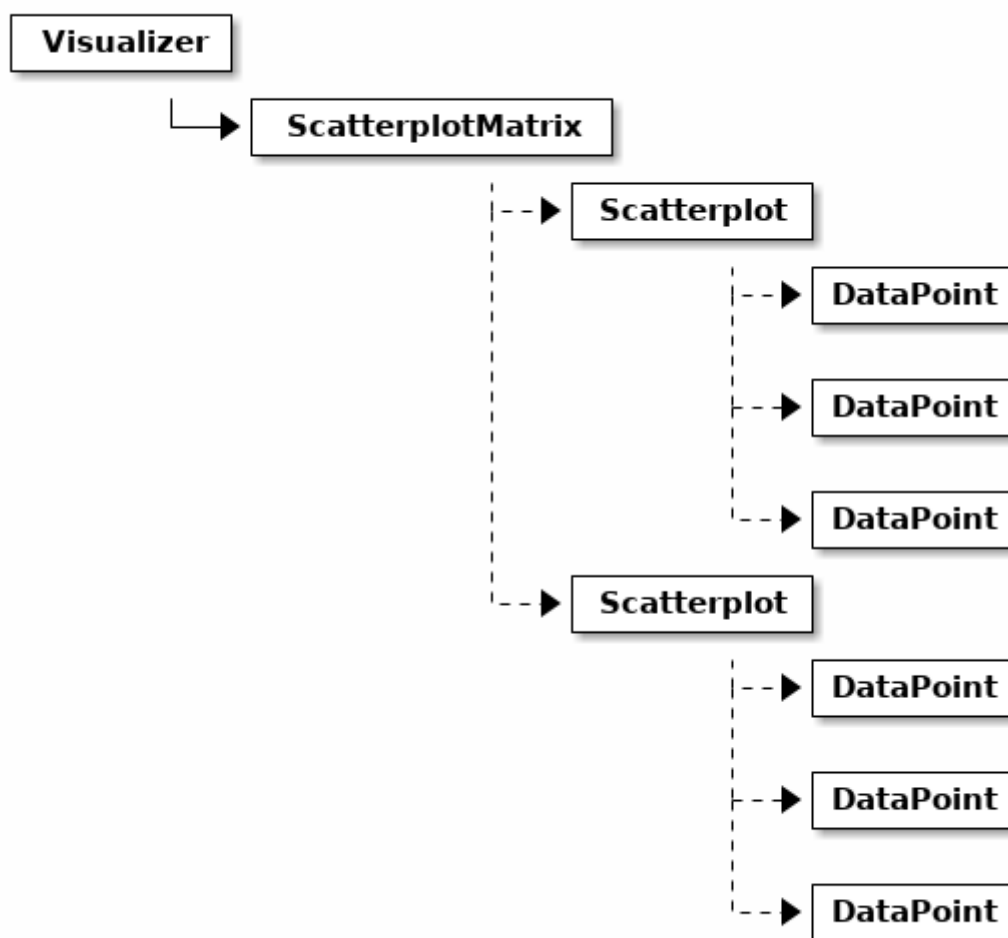


Abbildung 1. Szene

Visualizer

Die Schnittstelle des *Visualizer* besteht aus drei Methoden und einem Attribut. Die Methode *LoadDataSource(TextAsset)* wird vom *ImportDialog* aufgerufen, wenn der Benutzer im DropDown-Menü eine neue CSV-Datei auswählt. Neben dem eigentlichen Laden der CSV-Datei werden zusätzlich alle möglichen 3er-Kombinationen der in der Datei enthaltenen Dimensionen gebildet.

Hierzu wird *CalculatePossibleScatterplots()* aufgerufen, welche ein zweidimensionales *int*-Array zurück gibt. Die erste Dimension dieses Arrays stellt dabei die einzelnen Kombinationen der Dimensionen der CSV-Datei dar. Die zweite Dimension hat die Größe 3 und besteht aus den drei Indizes der jeweiligen Dimension der CSV-Datei. Der *ImportDialog* erhält die 3er-Kombinationen als *string* -Array über die Methode *GetPossibleScatterplots()*. Somit können dem Benutzer die möglichen *Scatterplot*-Varianten angezeigt werden. Wird der *Importieren*-Button angeklickt, ruft der *ImportDialog* die Methode *CreateScatterplotMatrix(int[])* auf. Über den Parameter werden die Indizes der vom Benutzer gewünschten *Scatterplots* übergeben. Es wird dann eine neue *ScatterplotMatrix* instantiiert. Wurde vorher bereits eine *ScatterplotMatrix* erzeugt, wird diese gelöscht. Es ist somit zu jeder Zeit immer nur eine *ScatterplotMatrix* in der Szene. Über das Attribut *pointSize* kann die Größe der *DataPoints* angepasst werden. Der *ImportDialog* inkrementiert bzw. dekrementiert diesen Wert, wenn der Benutzer die entsprechenden Buttons klickt.

Programmablauf

Das folgende Diagramm stellt den Programmablauf beim erstmaligen Starten der Anwendung dar:

- Blub
 - Blub
- blub

Blub

1. Ich
 - a. Mag
 - i. Züge
2. du
 - a. magst
 - i. LKW
 - A. Tut Tut

Neue Subüberschrift

Blub [Mein Bereich](#)

Beispiel 1. Mein Bereich

Übernimmt Ascii Zeug

Listing 1. Source Code

```
public void Foo(){  
  
}
```

Und noch Ref auf Überschrift [Neue Subüberschrift](#).

Quellen / Ressourcen

- [MiddleVR]: <https://www.middlevr.com/home>
- [IATK] Immersive Analytics Toolkit: <https://github.com/MaximeCordeil/IATK>
- [VRTK] Virtual Reality Toolkit: <https://vrtoolkit.readme.io>
- [SteamVR]: <https://steamcommunity.com/steamvr>
- [GitHub-Repository] Projekt Repository: <https://github.com/MHaselmaier/VR-Visualization.git>
- [Unity-Dokumentation]: <https://docs.unity3d.com/Manual/index.html>
- [MiddleVR-Dokumentation]: <https://www.middlevr.com/doc/current/>