Blender Import/Export Addon

Inhaltsverzeichnis

- 1 Einleitung
- 2 Installation
- 3 Einstellungen
  - 3.1 Allgemein
  - 3.2 Import
  - 3.3 Export
- 4 Objektstrukturen in Blender
- 5 Materialien und Texturen
- 6 Plugins
  - 6.1 Animations and Events
  - 6.2 Transport Network Provider (Pfade)
  - 6.3 Custom Metadata Provider (Metadaten für Models)
  - 6.4 Materials and MatConfigs

1 Einleitung

Aufgrund der verschiedenen Import/Export-Möglichkeiten des Blender Addons, gibt es dort mehrere Einstellungen zur Auswahl, die hier erklärt werden sollen.
Außerdem bietet das Addon, zusätzlich zum reinen Mesh-Export, einige Funktionen um das Erstellen kompletter Fahrzeug Mods zu erleichtern. Damit diese Funktionen genutzt werden können müssen die Objekte in Blender jedoch eine bestimmte Struktur haben, die hier ebenfalls beschrieben werden soll.

2 Installation

Die jeweils aktuellste Version kann immer hier heruntergeladen werden:

Blender Import/Export Addon Transport Fever

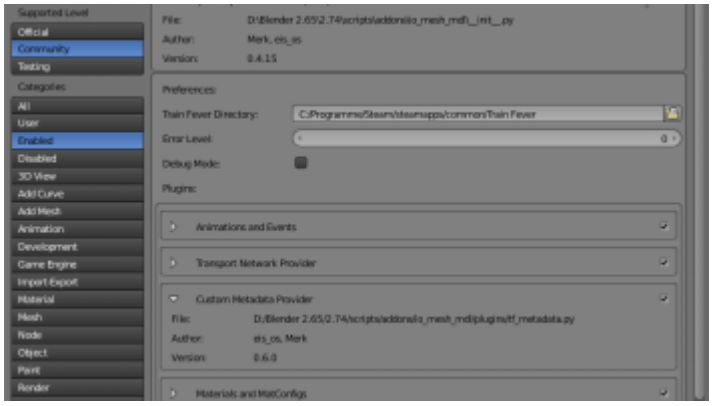
Blender Import/Export Addon Train Fever

Nach dem Download in Blender die "User Preferences" öffnen, zum Tab "Add-ons" wechseln, auf "Install from File" klicken und dort das komplette .zip-Archiv auswählen. (Oder manuell: Inhalt des .zip-Archivs in den Addons-Ordner von Blender (`Blender/<version>/scripts/addons`) verschieben.)
Danach muss das Addon in beiden Fällen noch aktiviert und die Einstellungen gespeichert werden.

3 Einstellungen

3.1 Allgemein

(seit Version 0.4.15)
Zu finden in den User Preferences (Tab "Add-ons", Kategorie "Import-Export", Detailansicht von "Train Fever (.mdl)").



Option	Funktion	Typ	Version
Train Fever Path	Pfad zu ausführbaren Datei von Train Fever, wird benötigt um zusätzliche Scripte zu laden, z.B. `laneutil` für Stationen und `vehicleutil` für Fahrzeuge. Bei der Steam Version auf Windows sollte er automatisch erkannt werden, kann aber immer auch manuell geändert werden. Nutzer anderer Version, bei denen die autmatische Erkennung nicht funktioniert, können sich gerne bei mir per PN melden.	String	
Error-Level	Gibt an welche Fehlermeldungen nach dem Import/Export direkt im Fenster angezeigt werden. (0: Nur solche durch die das Modell nicht korrekt importiert/exportiert wurde oder die wahrscheinlich dazu führen, dass das Spiel abstürzt. 1: Auch Fehler die das Spiel und das Addon selbst korrigieren können und Hinweise auf unlogische Daten)	Integer	
Debug Mode	Ist dieser aktiv, gibt das Addon in der Konsole relativ viele Statusmeldungen aus, um bei einem Fehler besser nachvollziehen zu können, wann dieser aufgetreten ist.	boolean [Ja/Nein]	

3.2 Import

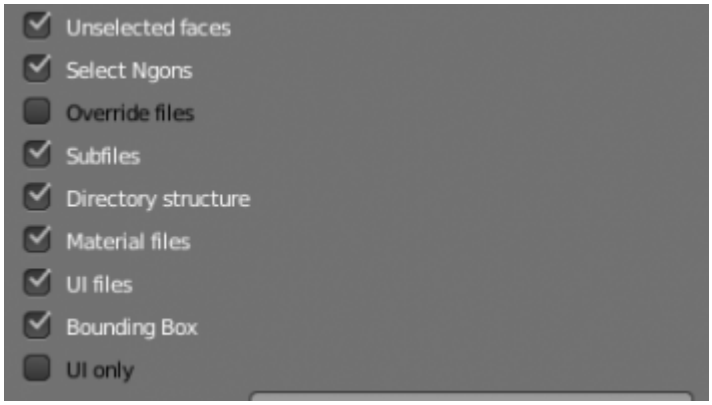
Der Importer bietet folgende Einstellungsmöglichkeiten:



Option	Dateitypen	Funktion	Typ	Version
Lod distance	.mdl	Gibt die Distanz an für welche das Lod Modell geladen werden soll (0 = alle Lods)	Integer	
UI-Only	.mdl	Ermöglicht es nur das UI-Icon für das Modell zu rendern, ohne es einzeln Importieren und Exportieren zu müssen	boolean [Ja/Nein]	ab 0.4.9
Material-Config	.grp, .msh	Gibt den Index der zu ladenden MatConfig an. Liegt der Wert außerhalb des Bereichs der gültigen Array-Indices, wird die letzte matConfig geladen.	Integer	ab 0.4.9
Flat-Faces		Gibt an ob auch Faces mit Shading: Flat erstellt werden sollen. Dies führt dazu, dass mehr Vertices und Edges zusammengefasst werden können.	boolean [Ja/Nein]	ab 0.4.12

3.3 Export

Der Exporter bietet folgende Einstellungsmöglichkeiten:



Option	Dateitypen	Funktion	Typ	Version
Hidden Faces		Versteckte Faces werden exportiert	boolean [Ja/Nein]	

Unselected Faces		Nicht ausgewählte Faces werden exportiert	boolean [Ja/Nein]	
Select Ngons		Flächen mit mehr als vier Ecken, die den Export verhindern, werden nach dem Exportversuch ausgewählt.	boolean [Ja/Nein]	
Override files		Bestehende Dateien beim Export überschreiben, die Hauptdatei (.mdl, .grp oder .msh) wird immer überschrieben, wenn nötig.	boolean [Ja/Nein]	
Subfiles		Alle für das Objekt benötigten Dateien erstellen (z.B. .grp und .msh bei Models)	boolean [Ja/Nein]	
Directory structure		Dateien in der selben Ordnerstruktur ablegen wie im "res"-Ordner des Spiels (Nein: alle Dateien werden im selben Ordner gespeichert)	boolean [Ja/Nein]	
Material files		Texturen und .mtl-Dateien erstellen	boolean [Ja/Nein]	
UI files	.mdl	UI-Grafiken für das Modell erzeugen (bis jetzt nur Seitenansichten von Fahrzeugen)	boolean [Ja/Nein]	
Bounding Box	.mdl	Die Bounding Box für das Modell aus den Meshdaten berechnen (Nein: der Wert der entsprechenden Custom Properties wird genutzt)	boolean [Ja/Nein]	
UI only	.mdl	Es wird nur das UI-Icon erzeugt und sonst keine Dateien. (seit Version 0.4.9)	boolean [Ja/Nein]	ab 0.4.9
Mesh dir		Name des Unterordners für die .msh Dateien (enthält das Eingabefeld keinen Text, kommen die Dateien direkt in das von "Model type" definierte Verzeichnis)	String	
Model type		Art des zu exportierendes Objektes, gibt den Pfad zu den Unterdateien an ("vehicle/train", "vehicle/waggon", etc.) Bei "Other" wird der unter **Custom model type** angegebene Pfad genutzt ("/" als Trennung der Ordnerebenen).	vordefinierte Strings	

Außerdem sind im Feld für den Dateinamen verschiedene Eingaben möglich:

- **leeres Feld**: der Dateiname wird automatisch generiert (Name des Ausgewählten Objektes + passende Dateieindung) und das ausgewählte Verzeichnis als res-Verzeichnis genutzt.
- **name.endung**: Der Dateiname entsprich "name", der Exporter sucht im Pfad nach einem "res"-Verzeichnis, wird keines gefunden, wird wieder das ausgewählte Verzeichnis als "res"-Verzeichnis genutzt.
    - endung = .mdl, .grp oder .msh: Die Dateiendung entspricht "endung".
    - sonst: Die Dateiendung wird automatisch generiert.

Die Standardwerte für die Dateinamen ergeben sich wie folgt:

- **Model**: Objektname
- **Group**: Alles was im Objektnamen vor dem ersten Punkt steht (enthält der Name keinen Punkt, wird der ganze Objektname genutzt)
- **Mesh**: Name des Meshes, nicht des Mesh Objektes!
- **Material**: Materialname
- **Textur**: Name der Grafik ohne Dateiendung

4 Objektstrukturen in Blender

Das Addon fügt neue Objektypen hinzu (Model, Lod, Group), die aber eigentlich alle Empties mit bestimmten Custom Properties sind. Diese Objekte können auf zwei Arten erzeugt werden:

- Über das Add-Menü der 3D Ansicht (Object Mode), Eintrag: Create TF Object.
- Indem einfach ein Empty erstellt wird und im Panel "Train Fever Object" unter "internal" der entsprechende Subtype ausgewählt wird ("Metadata", "Particle Emitter", "Capacity" und "Engine" existieren dort nur noch aus Kompatibilitätsgründen und sollten nicht mehr genutzt werden).

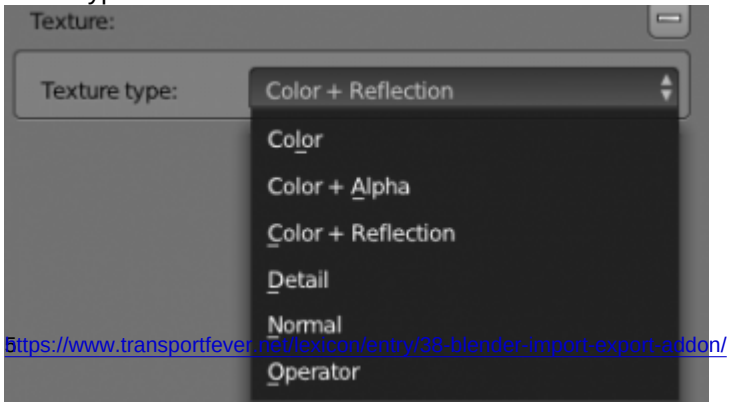
Um beim alle Funktionen nutzen zu können, müssen die meisten Objekte bestimmte andere Objekttypen als Parent haben. Eine solche Parent-Child-Verbindung, kann auch wieder auf zwei Arten erstellt werden:

- In der 3D-Ansicht (Objekt-Mode) zuerst das Child-Objekt und danach zusätzlich das Parent-Objekt auswählen. Anschließend die Tastenkombination Strg + P nutzen und eine der beiden Möglichkeiten auswählen (in der Regel ist wohl `Object (Keep Transform)`, die sinnvollere Variante)
- Das Child-Objekt auswählen und im Properties-Fenster, im Panel "Relations", bei Parent das entsprechende Parent-Objekt aus der Dropdownliste aussuchen.

Objekt	**erlaubte Parents**	**Ort des Einstellungs-Panels**
Model	keine	Object-Ansicht des Properties-Fensters
Lod	Model	Object-Ansicht des Properties-Fensters
Group	Model, Lod, Group	Object-Ansicht des Properties-Fensters
Mesh	Model, Lod, Group	Data-Ansicht des Properties-Fensters
Transport Network Provider	Model	Object-Ansicht des Properties-Fensters
Transport Network Group	Transport Network Provider	Object-Ansicht des Properties-Fensters

5 Materialien und Texturen

Die Texturslots müssen seit Version 0.4.12 nicht mehr nach einem bestimmten Schema benannt werden, stattdessen gibt es auch hier ein neues Panel (in der Texturansicht des Properties-Fensters) um den Texturtyp einstellen zu können:



[5] https://www.transportfever.net/lexicon/entry/38-blender-import-export-addon/

Ist der Texturslot nach dem vorher nötigen System benannt, wird automatisch der passende Texturtyp ausgewählt.

6 Plugins

6.1 Animations and Events

Datei	Ersteller	Panel
tf_anims.py	Merk	Train Fever Object

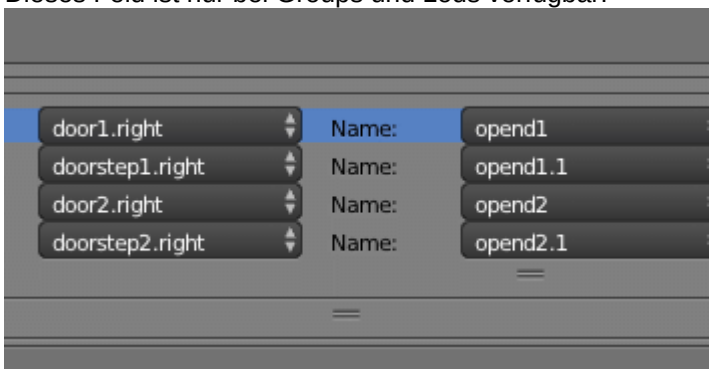
Ermöglicht den Export von Animationen und Events.

Animations:
Dieses Feld ist nur bei Meshes, Groups und Lods verfügbar und dient dazu einem Objekt mehr als eine Animation zuzuweisen:



Jeder Listeneintrag hat nur ein Auswahlfeld, in dem alle verfügbaren Animationen aufgelistet sind. Die hier ausgewählten Animationen werden dem Objekt beim Export zusätzlich zu seiner aktiven Animation zugewiesen, allerdings ist eine aktive Animation nicht zwingen erforderlich, man kann also auch nur mit diesem Feld arbeiten.

Events:
Dieses Feld ist nur bei Groups und Lods verfügbar.



Die einzelnen Events sind wieder Listeneinträge in diesem Feld und bestehen aus einen Namen (bei Groups frei wählbar, bei Lods prizipiell auch, allerdings wird Train Fever nur bestimmte Namen erkennen und nutzen) und einer Liste mit beliebig vielen Einträgen. Jeder Eintrag in dieser Liste enthält ein Auswahlfeld für das Child-Objekt dessen Event oder Animation aufgerufen werden soll und eine Auswahlfeld mit dem bestimmt

wird, welches Event/welche Animation aufgerufen werden soll. Ist `forward` nicht aktiviert wird die Animation/das Event rückwärts abgespielt. Beim Auswahlfeld für das Child-Objekt kann auch das aktuelle Objekt ausgewählt werden, wodurch beim Event zusätzlich auch noch eine Animation dieses Objektes ausgeführt werden kann.

Beispiel:
Für eine Drehfalttür wird eine Gruppe erstellt die beide Türteile enthält. Diese Gruppe eine Animation durch die sie um die z-Achse rotiert wird, zusätzlich bekommt der zweite Türflügel noch eine andere Animation, die ihn in die entgegengesetzte Richtung rotiert.
In der Gruppe wird nun ein Event zum Türöffnen erstellt, als Name wählt man z.B. "open_doors", in der zugehörigen Liste müssen dann zwei Einträge erstellt werden:

- Child: Tür-Gruppe; Name: Name der Animation der Gruppe
- Child: zweiter Türflügel; Name: Name der Türanimation

6.2 Transport Network Provider (Pfade)

Datei	Ersteller	Panel
tf_transportNetwork.py	eis_os	Train Fever TransportNetwork

Hier bin ich ehrlich gesagt selbst kein Experte, eis_os hat das denke ich aber sehr intuitiv gestaltet. Zunächst muss ein "Transport Network Provider"-Objekt erstellt werden (3D-Ansicht: "Add" -> "Create TF Object"), dem als Parent das Model-Objekt zugewiesen wird (Properties Fenster: "Object" -> "Relations" -> "Parent"). Danach können beliebig viele "Transport Network Groups" hinzugefügt werden, deren Parent wiederum der Transport Network Provider sein muss. Ist eine Transport Network Group ausgewählt, kann im Object-Tab des Properties Fensters beim Panel "Train Fever TransportNetwork" (ganz rechts) ausgewählt werden, für welche "Fahrzeugart" die Pfade in der Gruppe gelten (Personen, Trucks, Trams, etc.).Die Pfade selbst werden durch Bezierkurven (standard Blender Objekt: "Add" -> "Curve" -> "Bezier") dargestellt (Parent: Transport Network Group). Auch hier gibt es wieder ein eigenes Panel in dem weitere Einstellungen getroffen werden können:

- Track width: Breite des Pfads, soweit ich weiß nur bei den Pfaden, die den Wartebereich für Personen und Güter bestimmen, relevant.
- vehicleNodeGroupA/B: Gibt an für welches Terminal der Start(A)- bzw. Endpunkt(B) des Pfades als vehicleNode gilt (Position an der der Mittelpunkt eines Fahrzeugs hält; -1 = kein Terminal)
- personNodeGroupA/B: Gibt an für welches Terminal der Start(A)- bzw. Endpunkt(B) des Pfades als personNode gilt (weiß ich leider selbst nicht so genau; -1 = kein Terminal)
- personEdgeGroup: Gibt an für welches Terminal dieser Pfad als personEdge gilt (Bereich in dem Personen/Güter warten; -1 = kein Terminal)

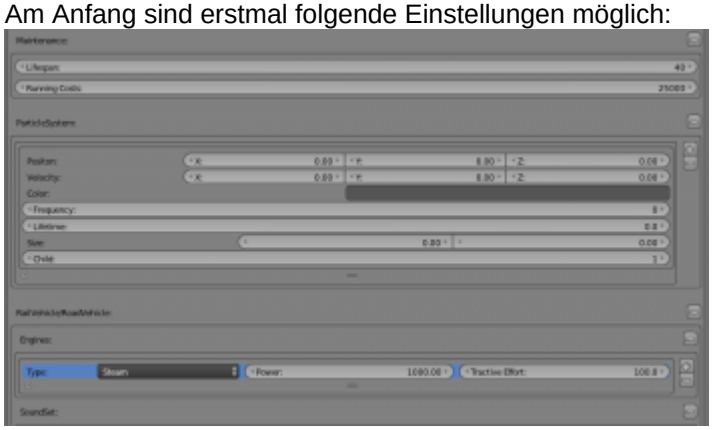
6.3 Custom Metadata Provider (Metadaten für Models)

Datei	Ersteller	Panel

tf_metadata.py eis_os & Merk Train Fever Object

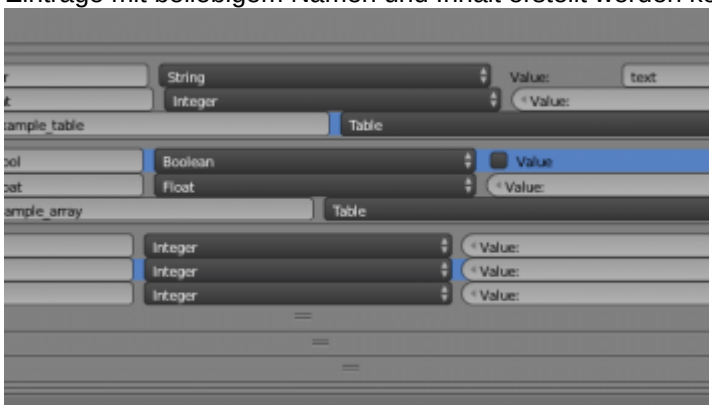
Dieses Plugin übernimmt fast alle Funktionen, die mit dem `metadata`-Eintrag in der. mdl zu tun haben, sollte es deaktiviert sein, werden die meisten Informationen daraus also weder Importiert noch Exportiert.

Am Anfang sind erstmal folgende Einstellungen möglich:



Prinzipiell haben die Felder die gleichen oder wenigstens ähnliche Namen wie die entsprechenden Felder in der .mdl. Bei Description wird der Inhalt der Felder `Name` und `Description` beim Export automatisch von `_(` und `)` umschlossen, damit die Einträge mit Hilfe der strings.lua auch übersetzt werden können. Außerdem können in dem Panel jetzt auch direkt die Werte für ParticleEmitter, Enignes und Capacities eingestellt werden, die jeweiligen Zusatzobjekte fallen ab Version 0.4.9 daher weg. Um bereits vorhande Daten von den Objekten in das neue System zu übertragen, müssen einfach in den jeweiligen Listen genauso viele Einträge erzeugt werden ("+"-Button), das Addon füllt sie dann automatisch mit den entsprechenden Daten. Die Objekte können danach auch gelöscht werden.
Der `Carrier`-Eintrag muss leider bei allen Objekten nochmal explizit gesetzt werden, da ich ihn leider nicht aus den vorhandenen Daten bestimmen konnte.

Sollten diese Möglichkeiten nicht ausreichen und noch weitere Einträge nötig sein, kann der Haken bei "Hide custom entries" entfernt werden, wodurch überall wo es möglich ist ein weiterer Unterpunkt erscheint, in dem Einträge mit beliebigem Namen und Inhalt erstellt werden können:



Dieses Beispiel würde folgenden Lua-Code erzeugen:

Code

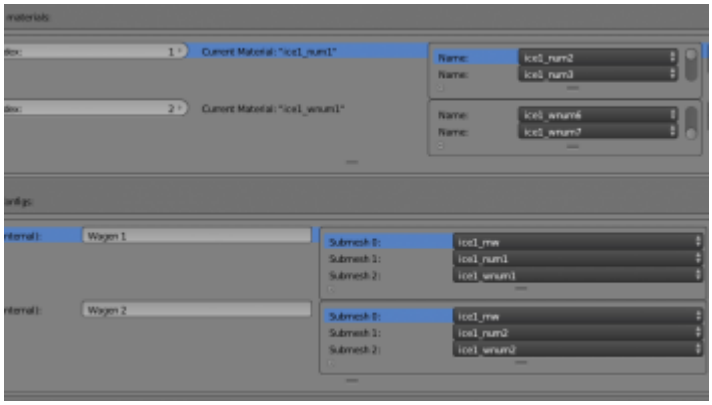
```
example_str                                    =                                    "test",
example_int                                       =                                    7,
example_table                                       =                                    {

                                                                example_array                =
},
```

6.4 Materials and MatConfigs

Datei	Ersteller	Panel
tf_materials.py	Merk	Train Fever Object

Dürfte nur für Leute interessant sein, die ein Mesh mit mehreren Materialien versehen wollen um es mehrfach zu nutzen (z.B. für verschiedene Lackierungsvarianten eines Fahrzeugs). Wenn man das nicht braucht, können die Einstellungen auch einfach ignoriert werden. ;) Image not found or type unknown



Die alternativen Materialien ("Additional materials") sind logischerweise nur für Meshes möglich und werden daher auch nur dort angezeigt. Mit den Buttons "+" und "-" am rechten Rand können Einträge erstellt bzw. gelöscht werden. Die Reihenfolge der Einträge ist irrelevant, allerdings sollte es nicht zwei Einträge mit dem selben `Slot Index` geben.
Dieser gibt an bei welchem Materialslot/Submehs die ausgewählten Materialien hinzugefügt werden sollen, als kleine Hilfe wird dahinter angezeigt, welches Material in diesem Slot gerade aktiv ist. Im Feld ganz rechts können beliebig viele Listeneinträge erzeugt werden, die jeweils ein Auswahlfeld bieten, in dem alle verfügbaren Materialien aufgelistet werden. Hierbei ist zu beachten, das bei allen Materialien die nicht direkt von einem Objekt genutzt werden, die Option "Save this datablock even if it has no users" aktiv sein sollte, da es sonst sein kann, dass sie nicht in der .blend-Datei gespeichert werden.

MatConfigs können für Meshes, Groups und Lods bestimmt werden. Auch hier werden die Einträge über die beiden Buttons ganz rechts erstellt/gelöscht, die Reihenfolge ist ebenfalls egal, wird allerdings beim Export beibehalten.
Das Feld für den Namen ist mehr oder weniger optional, da es nur im Auswahlfeld der Matconfig des Parent-Objektes genutzt wird (dort wo im Bild z.B. "ice1_mw" steht), für den Export ist es nur wichtig, dass der Name innerhalb des Objektes nur einmal vorkommt. Im Feld dahinter kann für jedes Submesh/Child-Objekt ein Material/eine MatConfig ausgewählt werden, die Anzahl der Listeneinträge richtet sich immer nach der Anzahl der Submeshes/Child-Objekte und kann nicht einzeln bearbeitet werden. Sollte durch bestimmte Änderungen (z.B. Löschen eines Child-Objektes) die Anzahl der Listeneinträge nicht mehr mit der benötigten Anzahl übereinstimmen, gibt das Addon einen Hinweis aus und bietet die Möglichkeit Einträge zu löschen oder hinzuzufügen.