

## **Spezifikation**

### **Manipulatorfahrzeuge - Physik**

**Matthias Liebl-Tabeling, Dipl-Ing. (FH)**

**09.02.2010**

Pearls Media Publishing Limited

Office London

Beamont House

172 Southgate Street

Gloucester, Gloucestershire, GL1 2EZ

## **INHALTSVERZEICHNIS**

Inhaltsverzeichnis .....	2
1 Allgemeines .....	2
1.1 Ziele des Dokuments .....	2
1.2 Mitgelieferte Virtools Basis CMO .....	2
2 Virtools CMO beschreibung .....	4
2.1 Virtools 3D Hierarchie .....	4
2.2 Virtools Scripte .....	6
2.3 Steuerung Fahrzeugantrieb .....	7
2.4 Steuerung Servoantriebe .....	7
3 Zielverhalten .....	8
3.1 Fahrverhalten .....	8
3.2 Manipulationsverhalten .....	8
4 Kontakt bei Rückfragen .....	9

## **1 ALLGEMEINES**

### **1.1 Ziele des Dokuments**

Das Konzept beschreibt auf der Grundlage

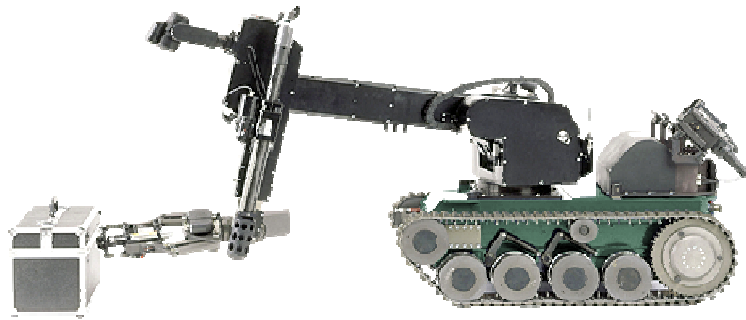
- der Forderung- und Stoffsammlung des Kunden und
- den Unterlagen über das Manipulatorfahrzeug „tEODor“

das Fahrverhalten des Fahrzeuges sowie die Manipulationseigenschaften der Arm- und Greiferstruktur in Hinblick auf eine möglichst realitätsgetreue Nachbildung.

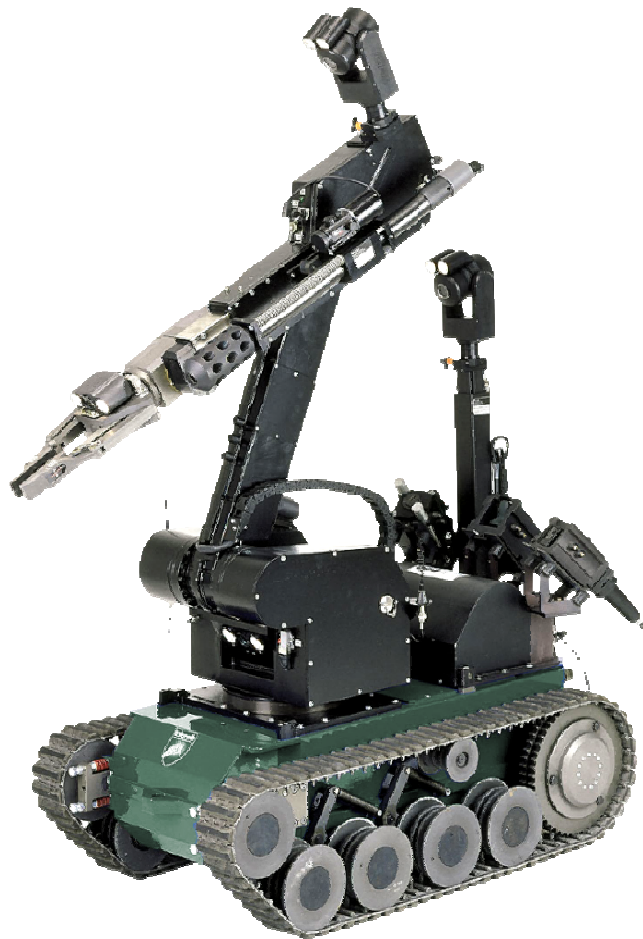
### **1.2 Mitgelieferte Virtools Basis CMO**

In der mitgelieferten Basisdatei „TEODOR\_PhysX\_Development.cmo“ ist das Manipulatorfahrzeug in Programmierung und Konfiguration soweit vorbereitet, dass eine Integration der PhysX Physics Engine möglichst ohne größeren Aufwand erfolgen kann.

Die Anwendung ist auf der Basis von Virtools 4.0 programmiert. Diese Version ist auch die Zielplattform.



*Abbildung 1: Manipulatorfahrzeug „tEODor“ im Einsatz*



*Abbildung 2: Manipulatorfahrzeug „tEODor“ in Fahrstellung*



Abbildung 3: Manipulatorfahrzeug „tEODor“ in Verpackfahrt in Virtools CMO

## 2 VIRTOOLS CMO BESCHREIBUNG

### 2.1 Virtools 3D Hierarchie

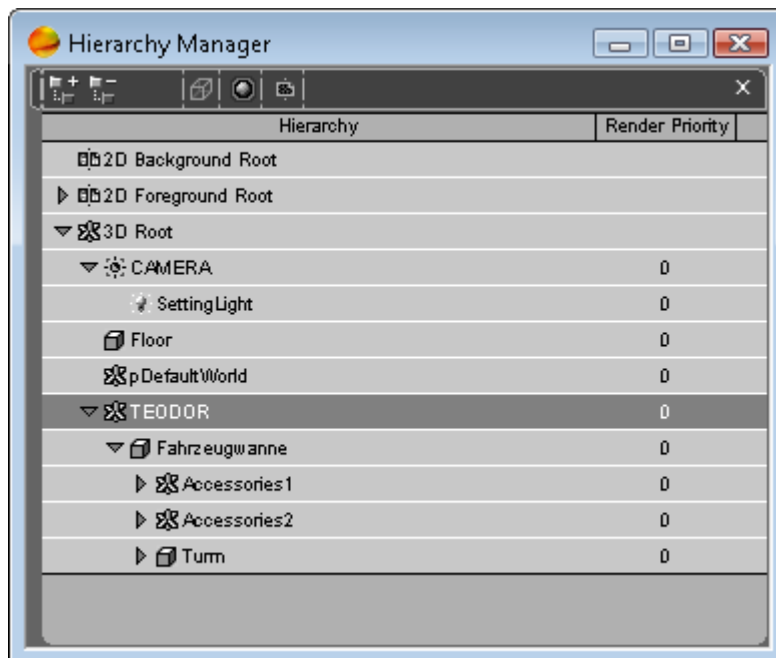


Abbildung 4: Virtools 3D Root des Manipulatorfahrzeuges „tEODor“

Wie man im Hierarchy Manager erkennen kann, ist das Main Parent des Manipulators das 3D Frame „TEODOR“. Hieran angegliedert ist dann die Fahrzeugwanne mit allen weiteren Hierarchien. Das 3D

Frame dient lediglich zum Setzen des Fahrzeuges bei der Initialisierung, wobei der Frame dann an Ort und Stelle verbleibt. Zum eigentlichen Fahren wird die Fahrzeugwanne als Main Object benutzt.

Neben dem Manipulator befinden sich in der Szene noch eine CAMERA mit angehängtem Settinglight (Directional Light), der Floor und der bekannte 3D Frame „pDefaultWorld“. Die CAMERA wird dann auch als aktive Kamera mit Orbit Funktion benutzt. Der Fokus der CAMERA wird neu gesetzt, indem man mit der Maus auf eine bestimmte Stelle im 3D Layout klickt.

**Zu den angehängten Hierarchien der Fahrzeugwanne ist zu sagen, dass vor allem die Hierarchie des Turmes wichtig sein wird, die PhysX in die Armhierarchie, welche im Turm ja beginnt, zu integrieren.**

Unter anderem befinden sich in den „Accessories2“ auch die Bestandteile der visuellen Fahrzeugkette. Wenn die CMO geladen wurde, sind die Kettenglieder noch nicht erzeugt worden. Sie werden nach dem Start erzeugt und an beiden Seiten an den 3D Curves „Chain\_curve\_L / \_R“ entlang angeordnet.

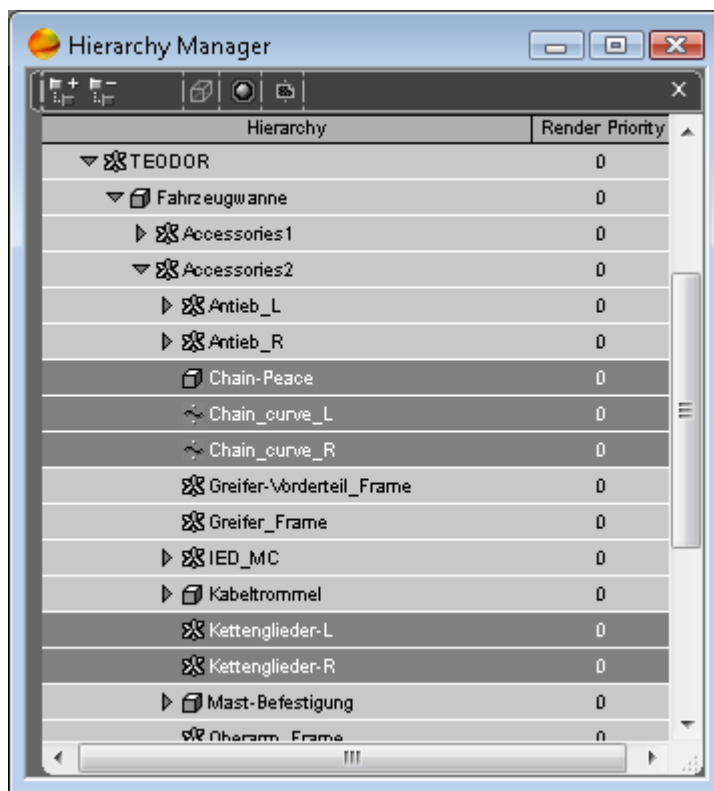


Abbildung 5: Virtools Bestandteile der visuellen Antriebsketten

Die 3D Frames „Kettenglieder-L / -R“ dienen zur hierarchischen Aufnahme der Kettenglieder. Es könnte angedacht werden, sie durch 3D Objekte zu ersetzen, welche zur PhysX gehören und die ebenfalls physikalisierten Kettenglieder an die Fahrzeugwanne koppeln, so dass im Falle einer Kettenbewegung sich das Fahrzeug entsprechend fortbewegt.

## 2.2 Virtools Scripte

Die PhysX muss im Wesentlichen in die Scripte (durch graue Farbe markiert dargestellt in Abb. 6), bzw. die Script Owner von der „Fahrzeugwanne“ bis zum „Greifer\_Vorderteil“ integriert werden.

Das „TEODOR START“ Script dient zur Initialisierung des Fahrzeugmoduls im eigentlichen Zielprogramm und ist hier temporär außer Betrieb. Nur die maßgebliche Aktivierung der entsprechenden Scripte plus Message an die Kameras wurde noch mal nachgesetzt.

Das „TEODOR Script“ sieht komplizierter aus, als es ist. Es dient zur Erzeugung der Steuerdaten des tEODors. Am unteren Ende des BB „Switch on Message“ befinden sich ein Behavior Graph (BG) „Control“. Dort drin werden die Antriebe für die Fahrzeugketten und die Steuerdaten der Servos für die Armhierarchie erzeugt. Mehr dazu in den Kapiteln „Steuerung Fahrzeugantrieb“ bzw. „Steuerung Servoantriebe“.

Unter dem BB „Switch on Message“ findet man einen BG „tEODor\_Interface“. Hier drin sind kaskadiert angeordnet alle Quellen der Steuer- und auch der Telemetriedaten des tEODors. Normalerweise gehören hier noch spezielle von der Firma szenaris GmbH programmierte Interface Elemente, die Daten zu einem realen Bediengerät austauschen. Sie sind nur an dieser Stelle nicht relevant. Daher wurden sie gelöscht.

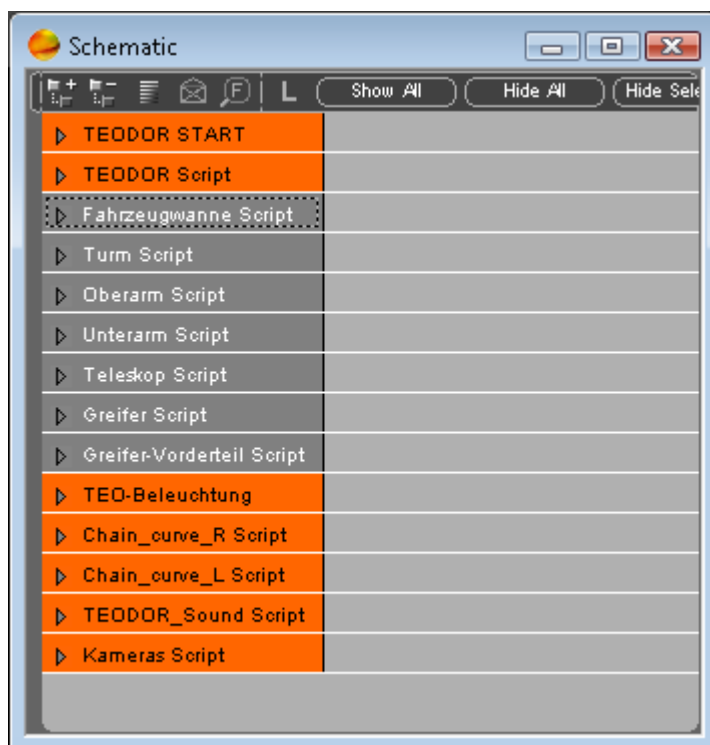


Abbildung 6: Virtools Schematic des Manipulator „tEODor“

Die Scripte „TEO-Beleuchtung“, „TEODOR\_Sound Script“ und „Kameras Script“ sind physikalisch ohne Bedeutung. Zuletzt findet man noch die Scripte „Chain\_curve\_R Script“ und „~\_R Script“, wo

die Antriebsketten erzeugt werden. Unter Umständen müsste hier die PhysX auch integriert werden.

## 2.3 Steuerung Fahrzeugantrieb

Im BG „Control“ im „TEODOR Script“ befindet sich ein weiterer BG „Emulate Fahren“. Mit den Keyboard Keys S, W, E und D wird hier der Kettenantrieb emuliert, wobei diese Antriebsdaten normal über das Interface vom Bediengerät kommen als Integers.

Die Buchstaben sind folgendermaßen belegt:

- S: linke Kette vorwärts
- W: linke Kette rückwärts
- E: rechte Kette vorwärts
- D: rechte Kette rückwärts

Im weiteren Programm werden diese Antriebsdaten auf einer Fahrzeugseite negiert verarbeitet, da es das Interface so hergibt. Daher an geeigneter Stelle bitte nicht wundern.

Verarbeitet werden die Antriebsdaten im „Chain\_curve\_R Script“ bzw. „Chain\_curve\_L Script“, wo die erzeugten Kettenglieder je nach Speed entlang der 3D Curves ständig neu gesetzt werden.

## 2.4 Steuerung Servoantriebe

Im BG „Control“ im „TEODOR Script“ befindet sich ein BB „Mouse Waiter“, welcher über einen angeschlossenen BB „2D Picking“ abfragt, ob nicht evtl. ein 2D Sprite vom späteren Frontend angeklickt worden ist. Wenn dieses nicht der Fall ist, wird ab dem angeklickten 3D Object so lange in der Hierarchie nach oben gesucht, bis ein im BB „Switch on Parameter“ relevantes Objekt gefunden wurde. Im angeschlossenen BG „Emulate Objects“ wird dann ein entsprechendes Steuerdatum bedient.

Das Steuerdatum an sich wird durch die Keyboard Keys „pos1“ und „end“ emuliert. Manipuliert wird immer das gleiche, nur durch die vorherige Auswertung, welches Armelement gemeint ist, wird es in das entsprechende Steuerdatum über BB „Identity“ zugewiesen. Die Verarbeitung der Daten geschieht in den entsprechenden Scripten der Armhierarchie.

Man braucht also im 3D Layout den tEODor nur irgendwo anklicken und der entsprechende Servo beginnt sich bei den Tasten „pos1“ und „end“ zu aktivieren. Im Moment ist es so, dass sich die Servos visuell alle bewegen, zusätzlich auch die Fahrzeugkette.

**Ziel:** Integration der PhysX in die bestehenden visuellen Funktionen. Die Armhierarchie soll also weiterhin mit herkömmlichen BB „Set Euler Orientation“ bzw. BB „Set Position“ durchgeführt werden. Die erwähnten BB sind bereits **ausgetauscht**, damit die Settings für die PhysX mit integriert sind.

**Vorsicht:** Die Euler Winkel werden komponentenweise eingegeben, wobei die nicht manipulierten Komponenten nicht immer Null sein müssen!

### **3 ZIELVERHALTEN**

#### **3.1 Zielplattform**

- Die Zielplattform ist Virtools 4.0 Dev.
- Wir haben einen eigenen VR-Player (nicht Webplayer) auf der 4.0 Basis.

#### **3.2 Fahrverhalten**

Folgende Fahreigenschaften soll der Manipulator haben:

- Kettenantrieb ist zu bevorzugen; unsichtbare Dummy Wheels nur als Fallback
- Geradeausfahrt
- Kurvenfahrt
- Wenden auf der Stelle (linke Kette vorwärts, rechte rückwärts)
- Treppen hinauf- und hinunterfahren, wobei der Kippunkt maßgebend ist; der Kippunkt an der Treppenkante kann durch Manipulation der Armhierarchie verändert werden (IED im Greifer kann dieses zusätzlich beeinflussen)
- Unterschiedliche Fahreigenschaften auf unterschiedlichen Untergründen (Beton, Gras, Schnee etc.) – Friction Abhängigkeit
- Zu hohe Fahrzeugneigung bei entsprechender Konstellation der Gelenke verursacht Umkippen des Fahrzeuges

#### **3.3 Manipulationsverhalten**

Folgende Manipulationseigenschaften soll der Manipulator haben:

- Verlagerung des Fahrzeugschwerpunktes je nach Manipulation der Armhierarchie, d.h. Einstellbarkeit der Gewichtsanteile der Hierarchieelemente
- Realitätsgetreue Kollisionsobjekte, welche in der Armhierarchie bereits als Children integriert sind, Prinzip der additiven Physik aus konvexen Objekten
- Möglichkeit, dass man sich die hintere Kamera auf dem Mast abrasieren kann, wenn der Arm zu weit nach hinten gefahren wird, oder wenn das Fahrzeug durch eine niedrige Unterführung fährt



- Im Greifer muss man etwas festhalten können (IED), wenn der Greifer geschlossen wird
  - Abhängigkeit von der Greifkraft, Eisenelemente brauchen eine höhere, Holz eine niedrigere Greifkraft
  - Zylinder etc. sollten sich in der Zange drehen, wenn sie am oberen Ende angefasst werden und die Zange sich nach unten neigt, da ja beim Zylinder nur zwei kleine Kontaktpunkte vorhanden sind
  - Greifer ist an dieser Stelle die Zange vorne am Greifer\_Vorderteil; Greifer in der Hierarchie ist das Parent vom Greifer\_Vorderteil
- Die Armhierarchie darf beim Fahren nicht zu stark wackeln, d.h. die Joints sollten sehr stark sein
- Das IED darf während des Fahrens nicht aus der Zange herausfallen, es sei denn die Greifkraft hat nicht ausgereicht
- Man muss mit dem Manipulator Türen öffnen können, d.h. die Klinke herunterdrücken und ziehen bzw. aufstoßen können
- Im Realgerät gibt es sogenannte „Rutschkupplungen“, die ein Betätigen der Gelenke von Hand zulassen, wenn man die Kraft hoch genug ansetzt. Dies verursacht auch ein Stehenbleiben der Gelenke im Betrieb, wenn der Kraftaufwand zu hoch wird, ein Gelenk zu bewegen. In unserem Fall müssten dann die Steuerscripte zum Setzen der Euler Angles aussetzen. Wäre schön, wenn man das mit berücksichtigen könnte.

## 4 KONTAKT BEI RÜCKFRAGEN

Matthias Liebl-Tabeling  
Dipl.-Ing., Senior Software Developer

szenaris GmbH  
Otto-Lilienthal-Str. 1  
D-28199 Bremen

Phone +49 [0]421 59647-37  
Fax +49 [0]421 59647-77  
matthias.liebl-tabeling@szenaris.com  
www.szenaris.com