# Virtual Reality im Cave – Projekt am ReachIn

Wolfgang Wögerer, David Füreder

## Das Ziel

Unser Ziel war es, ein Programm am ReachIn zu schreiben, bei der ein Benutzer an einem Objekt schnitzen kann. Um den gewünschten Effekt zu erzielen, werden viele kleine Würfel erzeugt und bei Berührung durch den Griffel gelöscht. Die Berührung ist natürlich fühlbar. An Abbildung 1 kann man die einzelnen Bausteine und den Griffel erkennen.

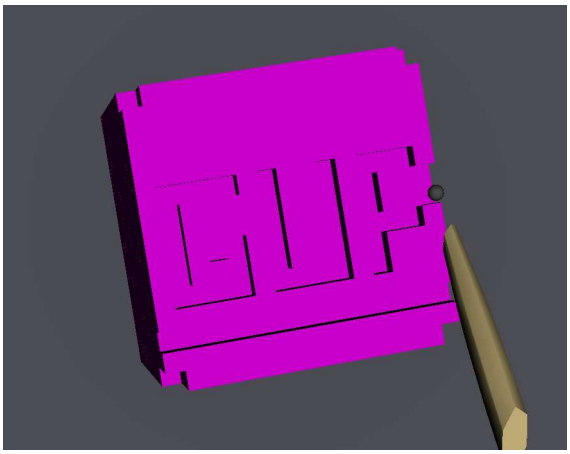
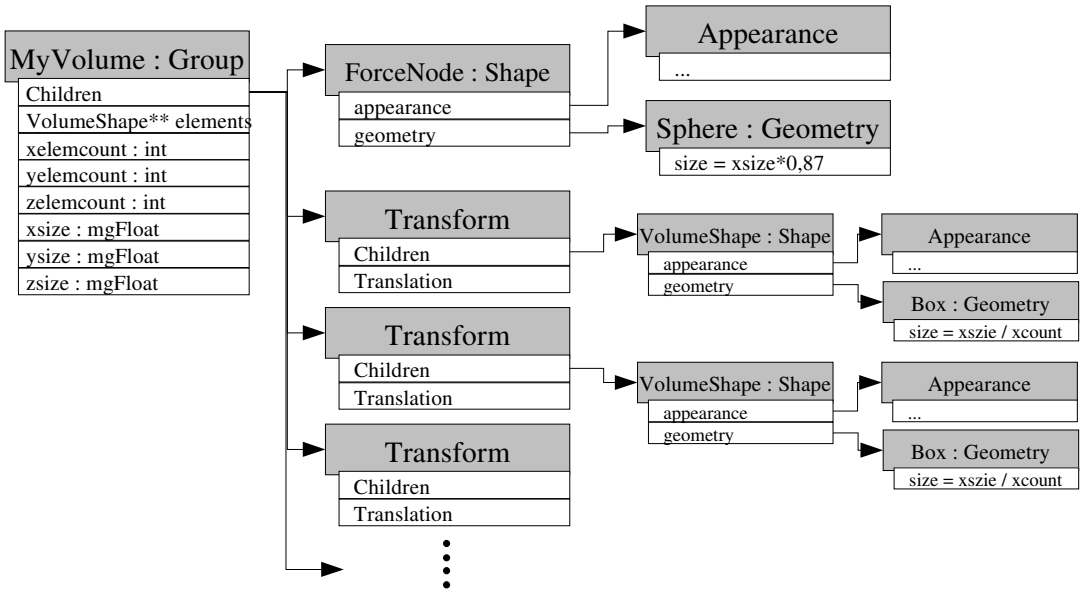


*Abbildung 1: GUP mit Griffel*

## Der Szenengraph

Die vielen kleinen Würfel werden duch die Klasse VolumeShape realisiert, welche im Knoten MyVolume gruppiert sind. MyVolume enthält ebenfalls noch eine durchsichtige Shape namens ForceNode. In Abbildung 2 fehlen der Übersichtlichkeit halber der Displayknoten sowie der Tranformationsknoten für MyVolume.




## Die Architektur

Um ein gutes Verständnis für die Architektur zu erhalten, reicht es den Aktionsverlauf vom Beginn

bis zur Löschung eines VolumeShapes zu verfolgen. Im Ausgangszustand ist MyVolume angelegt und der Griffel befindet sich irgendwo ausserhalb von MyVolume.

Der Griffel nähert sich MyVolume und trifft dabei zuerst auf ForceNode.

ForceNode ist eine Kugel, welche sich rund um die VolumeShapes befindet. Sie ist durchsichtig und besitzt keine Oberfläche und dient allein dem Zweck die Kollisionserkennung zu starten.

Sowie der Griffel nun auf ForceNode trifft beziehungsweise in ForceNode eindringt, wird geprüft, ob der Griffel ein VolumeShape berührt. Wenn eines berührt wird, wird es gelöscht. Für alle nicht gelöschten VolumeShapes wird eine Kraft abhängig von der Distanz zu Griffel berechnet und am Griffel angelegt.

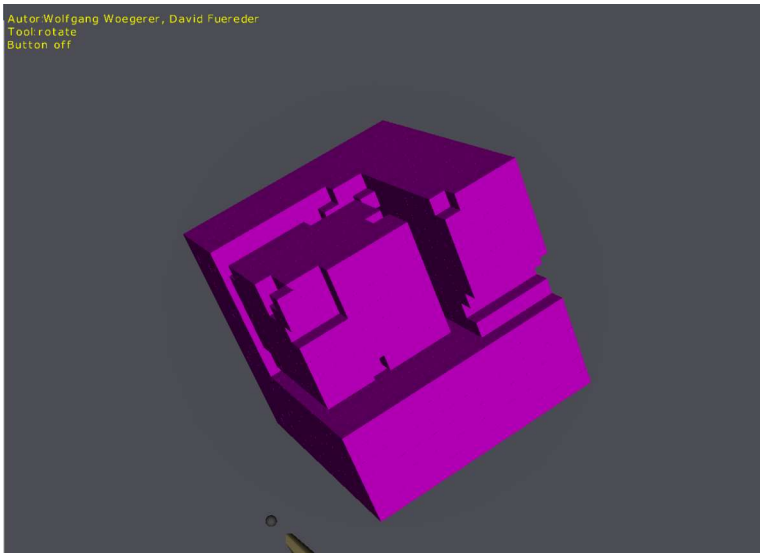



*Abbildung 2: Würfel aus Würfel geschnitzt*

## Probleme

ReachIn-Programmierung erfordert eine Menge Geduld. Die größten Schwierigkeiten lagen nicht etwa an der Programmierung selbst, sondern an:

- Abstürzen

  Das ReachIn stürzt so ziemlich bei allen auftretenden, manchmal von uns gemachten, Fehlern ab und kann dabei gelegentlich auch den Rechner neu starten.

- schlechter bzw. nicht vorhandener Fehlerausgabe

  Die Errorlogs beinhalten keine konstruktiven oder für uns verständliche Informationen über den eingetreten Fehlerzustand. Dies erschwert Debuggingvorgänge erheblich.

- unerklärliche Phänomene im Szenengraphen

  Oftmals war die Reihenfolge von Initialisierungen oder Einfügeoperationen in den Szenengraphen entscheidend, z.B. funktionierte die Transparenz von Objekten erst beim Einfügen in bestimmter Reihenfolge korrekt.

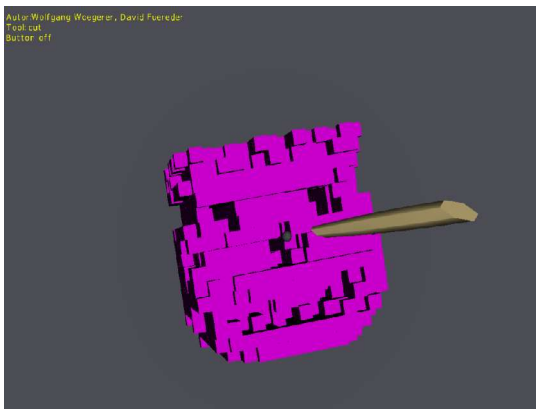



*Abbildung 3: Smile!*