

## **Dokumentation Team Driven**

Hupfer Daniel            935/0526989 e0526989@student.tuwien.ac.at  
Pacher Christoph        935/0525833 e0525833@student.tuwien.ac.at

Im Rahmen der Virtual Reality Labor Übung entwickelten wir eine Virtual Showcase Applikation, in der ein CAL3D Character Informationen über ein echtes 1:18 Automodell präsentiert. Das Automodell wird auch dreidimensional dargestellt, um es von allen Seiten betrachten zu können kann es näher herangeholt werden. Der Benutzer wird über getrackte Shutterglases ein stereoskopischer Blick auf die virtuellen Elemente Des Showcase ermöglicht. In dieser Applikation haben wir folgende Shader implementieren:

- Per Pixel Blinn Lighting (auch wenn der Shader im Code unter Phongshading zu finden ist)
- Per Vertex Normal Environment Fresnel Reflections
- Dot3 Bump Mapping
- Fresnel Reflective Dot3 Bump Mapping

Zu Begin haben wir versucht unsere Ideen in der Studierstube umzusetzen, stießen jedoch bald an Grenzen, die uns dazu bewegten eine eigene Applikation zu entwickeln. Dadurch mussten wir die in der Studierstube bereits vorhandenen Funktionen wie z.B. Trackeranbindung und Off Axis Camera selbst implementieren, wodurch uns zu wenig Zeit für Shadow Mapping blieb. Da diese Applikation nur im Showcase Sinn macht, beinhaltet unsere Abgabe nur unseren Cal3d Modellviewer samt CG Shadern.

Angelehnt an den offiziellen Cal3d Viewer beinhaltet dieser eine Eigenentwicklung von Klassen, die die Cal3D Modeldaten managen und darstellen und mit folgenden Zusatzfeatures ausgestattet ist:

- Erweiterte Anbindung an 3DS Max. Einem Submesh kann in 3D Max ein CG Shader durch eine spezielle Endung im Meshnamen zugewiesen werden. Ohne größere Änderungen am Cal3d Core ist es leider nicht möglich diese Informationen über die 3DS Max Materials zu transportieren
- Anbindung an den von uns Entwickelten ShaderManager. Material und Shaderinfos werden beim Rendern der Submeshes an den Shadermanager übergeben.
- Der ShaderManager verwaltet alle Shader sowie die Informationen, die zur Darstellung dieser notwendig sind. So ist es möglich schnell den Shader beim Rendern eines Submeshes mit einem Funktionsaufruf zu wechseln. Der ShaderManager übernimmt das updaten der nötigen Matrizen und initialisiert den CG Shader.
- Optimierte Versionen für statische und animierte Modelle. Bei beiden werden die Geometriedaten über die Vertex Buffer Object Extension auf den Grafikkartenspeicher ausgelagert. So ist es möglich den praktischen Cal3d Exporter auch für statische Objekte zu verwenden, ohne größeren Overhead beim Rendern

## **Quellen:**

Cal3d und dessen offizielle Viewer Implementation. Diese diente als Ausgangspunkt für unsere Implementation, was oft nicht leicht war, da die Cal3d Dokumentation entweder veraltete oder erste gar nicht vorhanden ist. Das Forum ist genauso wenig hilfreich, da so gut wie nicht frequentiert. So verbringt man z.B. schon mal eine Woche beim Aufspüren eines

Fehlers im Cal3d Code, der mit einer Doku oder Hilfe aus dem Forum schnell behoben gewesen wäre.

GameDev/NeHe Tutorials als Ausgangspunkt für die VBO Extension

GameDev/NeHe, CG Toolkit, „The CG Tutorial“ für CG ansich sowie Bump Mapping und Environment Reflective BM

### **Starten des Viewers:**

Den mitgelieferten dependencies Ordner auf C:\ kopieren und eine System Path Variable darauf setzen. VS Solution starten.

Oder die bereits kompilierte Version aus dem Debug Ordner samt den \*.cg Files und den Ordnern „models“ und „cubemaps“ in den „bin“ Ordner der dependencies kopieren.

### **Interaktion im Viewer:**

Mouse Rotation	View Rotation
w	move in
s	move out
e	environment an/aus
m	environments wechseln
f	Licht mit Maus bewegen an/aus

### **Verschiedene Car Models:**

Da ein Eingriff in den automatischen Renderingprozess mittlerweile nur mehr sehr umständlich zu realisieren ist, haben wir das Automodell in 3 Versionen mitgeliefert.

1. Ohne Environment Reflection BM
2. Eine Version bei der die Front der Karosserie in 3DS Max auf 30% der ursprünglichen Vertex Anzahl reduziert wurde und aus dem Vergleich Low/Highpoly Mesh eine Bump Map generiert wurde.
3. Zur Verdeutlichung des ER BM eine Version bei der auf der ganzen Karosserie eine „bumpy“ Bump Map gelegt wurde

Ursprünglich hatten wir vor, für alle Meshes wie bei Modell 2. Bump Maps zu generieren. Es ist jedoch schwierig und sehr aufwendig wirklich gute Bumps zu generieren und vor allem bei den Reflexionen fällt jeder Fehler auf. Da in Cal3d ein Mesh und Material Name nur einmal vorkommen darf, und die exportieren Modelle alle aus der selben 3D Max Scene kommen, können diese nicht gleichzeitig geladen werden. Zum Wechseln der Modelle muss daher in der main.cpp gleich am Anfang des Files die globale Variabel carNum verändert werden.

### **Hardware Anforderungen**

Erstellt auf einem A64 2.Ghz, Radeon 9800 System, mit ca. 100 Fps bei Viewer Start.

Erfolgreich auf Rapunzel getestet.