

Vulkan – The Future of High Performance 3D Rendering

Studiengang: BSc in Informatik | Vertiefung: Computer Perception and Virtual Reality
Betreuer: Prof. Marcus Hudritsch
Experte: Dr. Harald Studer (Optimo Medical AG)

Vulkan ist eine Programmier-Schnittstelle für 3D-Computergrafik sowie Berechnungen auf Vulkan-fähigen Geräten. Aufgrund der steigenden Anforderungen an die Computerhardware, soll Vulkan eine Alternative zu bereits bestehenden Lösungen bieten. Auf Kosten der Entwicklungszeit und erhöhter Verantwortung an die Anwender soll Vulkan die Ressourcen der heutigen Hardware besser balancieren bzw. optimaler benutzen können.

Ziel

Die Ziele dieser Arbeit waren es einen Abstraktions-Layer um die Vulkan API und OpenGL API zu bauen. Dies soll dem Benutzer erlauben eine Grafik-szene ohne genaueres Wissen über die verwendete API zu bauen. Zusätzlich soll ein Performance-Vergleich der beiden APIs die Vorteile der neuen Programmier-schnittstelle aufzuzeigen.

Abstraktionslayer

Der Abstraktionsgrad der Vulkan Programmierschnittstelle ist viel geringer als bei OpenGL. Dies soll dem Benutzer mehr Verantwortung über die Verwaltung der Hardwareressourcen geben und ihm ermöglichen diese für seine Applikation optimaler zu nutzen. Dieser geringere Abstraktionsgrad führte zu einer Schwierigkeit bei der Aufgabe, da der OpenGL Treiber im Gegensatz zu vulkan viele Aufgaben übernimmt. Die Funktionalitäten der Schnittstellen wurden in ein Interface vereinheitlicht und für jede Schnittstelle umgesetzt. Da der OpenGL Treiber viele Aufgaben automatisch vornimmt, wurden einige Funktionen des Interfaces im Falle von OpenGL leergelassen.

Szenengraph

Ein Szenengraph ist eine Darstellungsart für eine Szene welche im Umfang dieser Arbeit gewählt wurde.

Dieser ermöglicht es dem Benutzer die Daten in einer Baumähnlichen Struktur darzustellen. Die einzelnen Knoten enthalten gespeicherte Transformationen welche die Koordinaten der Nachfolgeknoten beeinflussen. Die Endknoten im Baum stellen ein darstellbares Objekt dar. Somit kann z. B. ein Mensch einfacher dargestellt werden, da die Rotationen seines Oberkörpers die Position der Arme durch einen gemeinsamen Knoten beeinflusst. Die Arme sind eine Ebene tiefer um die individuelle Transformation der Arme darzustellen.

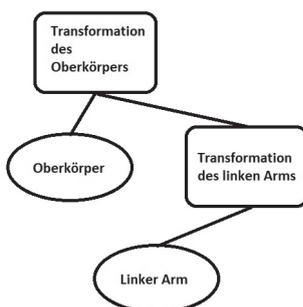


Casimir Benjamin Platzer

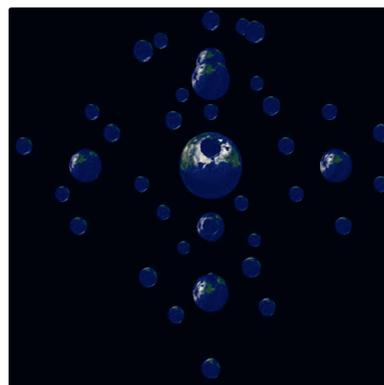
Massenszene

Für den Performance-Vergleich wurde eine Massenszene erstellt. Diese besteht aus mehreren texturierten Kugeln welche rekursiv gebaut wird. Obwohl jedes Objekt eine Kugel ist, die Texturen und das Shader-Programm identisch sind, wurde angenommen, dass jedes Objekt eine individuelle Form, Textur und Shader-Programm hat. Mit anderen Worten: Der Rechenaufwand mit verschiedenen Formen, Texturen und Shaderprogrammen ist vergleichbar mit der Massenszene. Aus einem Performance-Vergleich ging hervor, dass der Rechenaufwand bei OpenGL höher ist. Dies wird an der Tatsache liegen, dass der Gebrauch mittels der Vulkan API für eine Applikation optimiert werden kann, in diesem Falle des Szenengraphen.

Szenengraph für einen Arm und Oberkörper



Beispiel eines einfachen Szenengraphen.



Massenszene mit rekursiv generierten Kugeln.