PhysX Evaluation

Softwarepraktikum Computergraphik

Daniel Brock, Robert Kreuzer, Simon Kufner

5. Juli 2010

PhysX im Umgang mit Dominosteinen, Strömungen und Steinblöcken

Eine Evaluation

- Erstellen einer Szene mit Dominosteinen zur Einführung in PhysX und dessen Festkörper-Simulation
- Erstellen einer Szene mit Partikelsystemen zur Simulation von Strömungen
- Umsetzen der akquirierten Erfahrung in eine komplexere PhysX-Anwendung
- Generierung einer Szene mit hochpolygonalen Steinen aus Banteay Chhmar

Motivation

- Erweiterung von Virtueller Realität durch Physik-Simulationen führt zu realistischeren und unterhaltsameren Anwendungen
- Implementierung jedoch aufwändig
- Hoher Bedarf an Rechenleistung und Speicher

Berechnung auf Grafikkarten

- Entlastung der CPU
- Viele physikalische Berechnungen lassen sich gut auf Vektorprozessoren (GPU) parallelisieren
- Jedoch Abhängigkeit von Hardware (Physik-Beschleuniger / Grafikkarten)
- Verkürzen von Entwicklungszeiten durch Middleware Physik Engines (Plattformunabhängigkeit)

Physik-Engine PhysX

- Initial von Ageia entwickelt
- Aufgekauft von NVIDIA und portiert auf CUDA
- HW-Unterstützung auf Grafikkarten von NVIDIA (ab GeForce 8 Serie, CUDA-fähig, min. 32 Recheneinheiten und 256 MB Grafikspeicher)
- SW-Emulation auf CPU mit geringerer Leistung möglich

Physik-Engine PhysX

- Unterstützte Plattformen sind Windows, Apple Mac OS X, Linux, Nintendo Wii, Sony PlayStation 3, Microsoft Xbox 360
- Engine und SDK sind unter http://www.NVIDIA.co.uk/object/physx_new_uk.html erhältlich.
- Binärdateien sind für kommerzielle und nicht-kommerzielle Nutzung kostenlos, Quelltext kostet $50k
- Physik-Simulation in Echtzeit
- Feste und weiche Körper, Flüssigkeiten, Gewebe, Kraftfelder, ...

Limitationen

- Höchsten 64k Objekte
- Höchstens 256 Polygone als konvexe Hülle
- Verschiedene Flüssigkeiten können nicht interagieren
- Gewebe kann nicht mit anderem Gewebe oder weichen Körpern interagieren
- Kollisionserkennung nicht zwischen allen Repräsentationen
- Physikalische Regeln verhindern u.U. eigenes Verhalten
- Keine HW-Unterstützung auf Linux, Mac OS

Dominosteine

- Test auf realistischen Domino Effekt
- Wovon hängt die Fallgeschwindigkeit ab?
- Größe der Dominos
- Zeitschritt
- Schwerkraft

- Drehung über Rotationsmatrizen oder Quaternionen
- Merkt sich globale Koordinaten und Ausrichtung
- Box-Objekt teil des SDK
- Kurvenradius abhängig von der Anzahl der Dominos
- Max Dominos systemabhängig (2000 - 5000)

Simulation von Strömungen und Gasen durch Partikelsysteme

- Erstellung von Partikeln (NxFluid) durch NxFluidEmitter oder durch Laden eines gespeicherten Zustands
- Entfernen durch Drains (Flag) oder durch Absterben
- Interaktion mit anderen registrierten Objekten (Ausnahme andere Partikelsystemen)
- Expliziter Programmieraufwand bei Meshes und beidseitigem Abstossen
- Simulation in 3 Modi möglich
  - Normaler (einfacher Modus) - Perfomant
  - Smoothed Particle Hydrodynamics - Kräfte zwischen Partikel werden berücksichtigt
  - Gemischter Modus - Dichte wird berücksichtigt

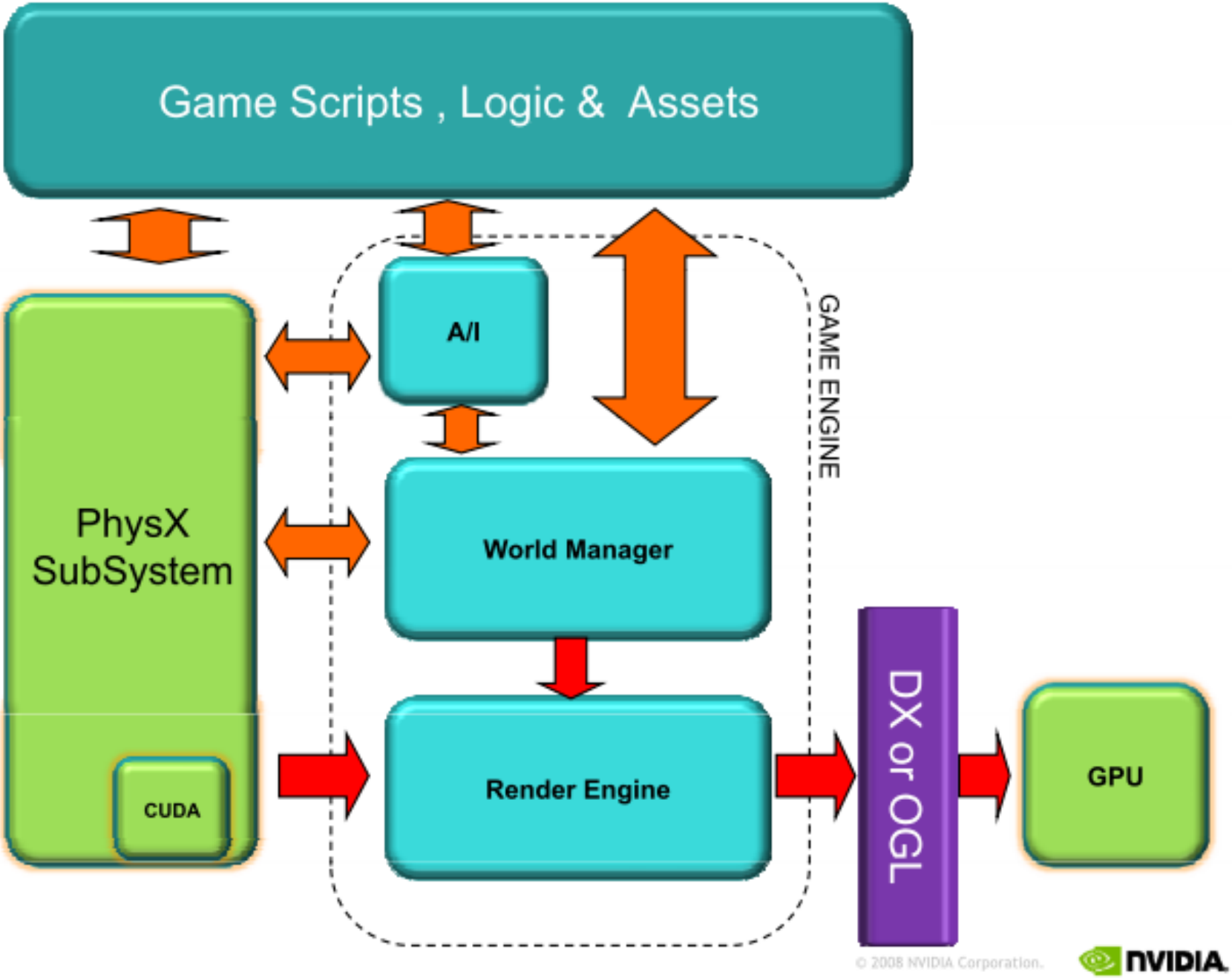
Steine von BanteayChhmar und konvexe Hüllen

- Hochauflösende 3D-Scans von Steinen
- Konvexe Hüllen für die Kollisionsabfrage berechnen
- ConvexDecomposition
  - extrem langsam, 3h für den kleinsten Stein
  - gute Näherung
  - viele nützliche Parameter
  - stürzt bei allen außer dem kleinsten Stein ab
- MeshLab
  - schnell
  - erzeugt viel zu große Meshes
  - keine Parameter

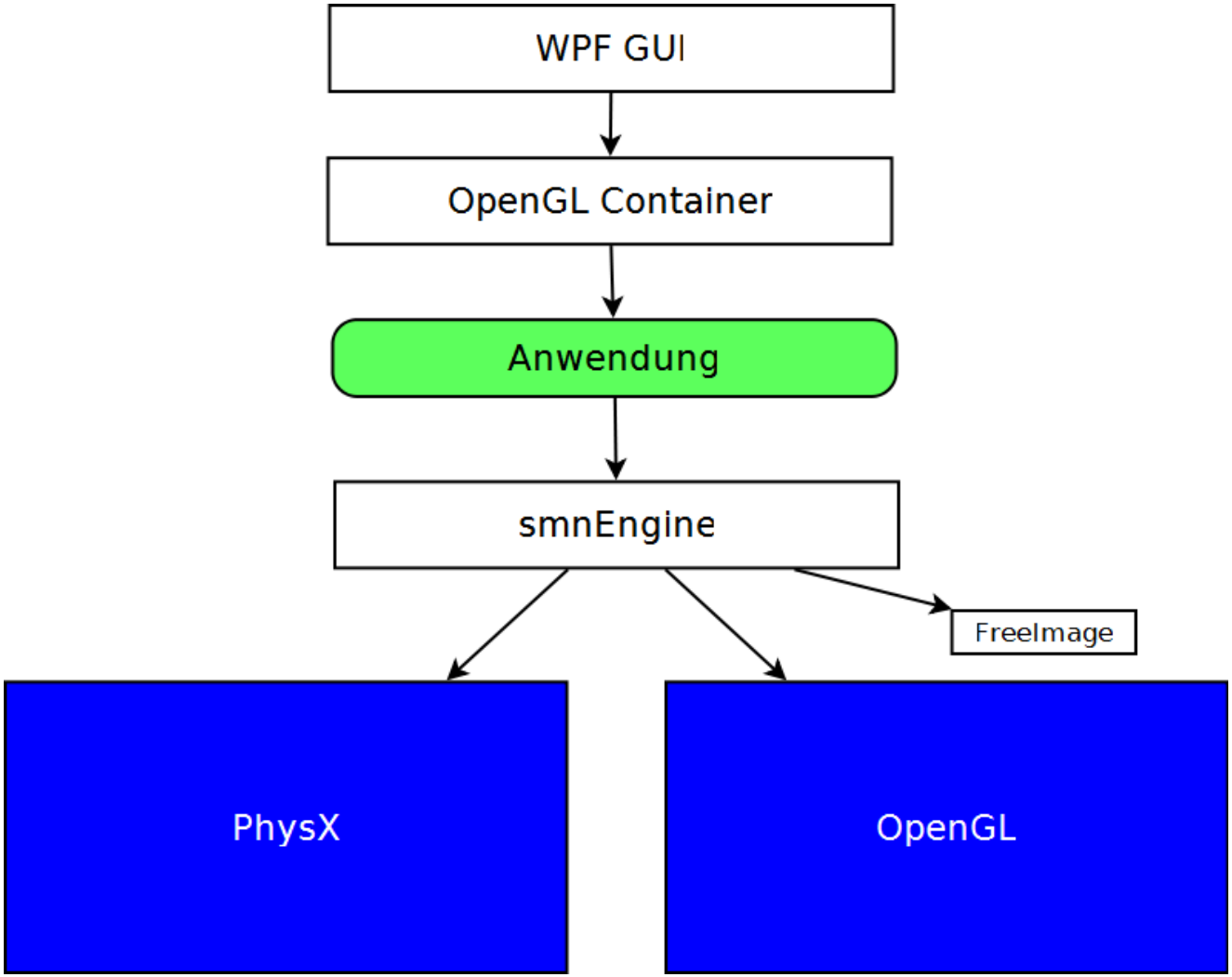
- DecomposeSample
  - schnell
  - viele Parameter
  - leider kein Parameter für maximale Anzahl Dreiecke, lediglich für maximale Anzahl Vertices
  - passende Parameter experimentell für jeden Stein einzeln von eigener Software ermitteln lassen

Verwendete Technologien

- NVIDIA PhysX
- Microsoft Visual Studio 2010 Ultimate
- Windows Presentation Foundation
- Visual C#, Managed C++, Visual C++, Marshalling
- FreeImage
- DecomposeSample, ConvexDecomposition, MeshLab


Game Scripts , Logic & Assets
A/I
GAME ENGINE
PhysX SubSystem
CUDA
World Manager
Render Engine
DX or OGL
GPU
© 2008 NVIDIA Corporation.
NVIDIA

Unsere Architektur



Zusammenfassung

- OpenGL im Microsoft Kontext schlecht unterstützt $\Rightarrow$ Verwenden von DirectX
- PhysX mächtig und leicht einzubinden
- PhysX eher für spielerische Anwendungen, nicht für exakte Simulationen
- Hochpolygonale Modelle benötigen Optimierungen (einfachere Kollisionsrepräsentationen)