



Entwurf und Programmierung einer Rendering Engine:
Übungsvorbesprechung

Robert F. Tobler

VRVis Research Center
Vienna, Austria



Bereitgestellte Entwicklungsumgebung

Programmierumgebung

- visual studio community: kostenlos von Microsoft
- letztes directx sdk:
<https://www.microsoft.com/en-us/download/details.aspx?id=6812>
- AardvarkLight: Gerüst eines Rendering Kernels in das Module hineingehängt werden können (kann verwendet werden, muss aber nicht)
<http://www.cg.tuwien.ac.at/courses/RendEng/>
- jeder verwendet seinen eigenen Rechner

Aardvark - Student

Einfaches Framework für Rendering

- Vektorbibliothek
- Command Queue
- Minimal Rendering Kernel
- verwendet DirectX9

Aardvark.Math 3D-Library

Vector-Library with primitive Data types

- all structs
- V3s, V3i, V3f, V3d
- M44f, M44d
- source code der Library: <https://github.com/vrvis/aardvark>

Aufgabenstellung

3 Teile

- Generierung von Geometrie
- Bearbeiten der Geometrie
- Rendern

Generierung von Geometrie

Einfache Demonstrations-Geometrie, z.B.

- einfache Drehkörper mit parametrisierter Detaillierung
 - Sphere, Zylinder, Cone, Hyperboloid, Paraboloid, ...
- einfache Terrains
 - random 1/f midpoint offset
- random triangles (z.B. für culling algorithms)
- 2d-Labyrinth, 3d-Labyrinth
- raster-basierte zufällig generierte Stadt
- ...

Optionen bei der Geometriegenerierung

Auf jeden Fall:

- Menge der generierten Geometrie

Optional, je nach Aufgabenstellung:

- Farben
- Normalen
- Texturen

Bearbeiten der Geometrie

Modul für Bearbeitung der Geometrie

- Quad-Tree/ Oct-tree für simple view-frustum-culling
- BSP Baum für transparente Darstellung
- k-D Baum für Intersection queries
- occlusion-culling-Datenstrukturen (2D für die Übung ausreichend)
- Subdivision Surfaces, Generierung von mehr Details
- Berechnung von Triangle-Strips für beschleunigte Darstellung (nicht im Aardvark-Framework)
- ...

Parametrisierung der Bearbeitung

Wichtigster Parameter: ON/OFF

- Vergleich vom Rendering der unbearbeiteten und der bearbeiteten Geometrie
 - z.B. mit und ohne View-Frustum culling

Weitere Parameter:

- je nach Modul
- Beispiele:
 - Unterteilungsgrad bei Quadtree
 - kd-Tree Parameter

Rendern

Sehr simples Modul

- Verpacken der übergebenen Daten für Rendering (erzeugen von VertexGeometry Objekten)
- Setzen der geeigneten Rendering-Parameter (DX9), Absetzen des Rendering Calls (im AardvarkLight framework, geschieht das durch erzeugen eines simplen Szenegraphs mit VertexGeometries)
- Interaktives Umschalten zwischen bearbeiteter und unbearbeiteter Geometrie

Datenaustausch zwischen den Modulen

Daten im Transponierten Ansatz

- möglichst viel mittels Arrays gleichartiger Primitivdatenstrukturen
- Module erhalten nur die Arrays die sie für die Bearbeitung benötigen

Ergebnis

- 2-Seitiger Bericht (bis 2 Tage vor der mündl. Prüfung)
 - gewählte Aufgabenstellung, gewählte Algorithmen
 - Ergebnisse: screenshots, Performancezahlen, ...
 - Analyse der Performance: z.B. Vergleich der Framerate bei Verwendung des Moduls für die Geometriebearbeitung bei verschiedenen Datenmengen und verschiedenen Parametern
- programmierte Module im Source Code
 - Executable mit Anleitung zur Ausführung wenn das Programm auf unserem Rechner hergezeigt werden soll (bitte das Executable in diesem Fall auch 2 Tage vorher mitschicken)
- Besprechung der Ausarbeitung bei der mündlichen Prüfung
- alles als ein zip-File an: rft@vrvis.at

Bemerkungen

Andere Frameworks dürfen verwendet werden:

- es gibt **keine Abzüge** für andere Implementierungen und Sprachen (Java, C, C++, Haskell, ...)
- das C# Framework wird zur Verfügung gestellt um die Aufgabe zu vereinfachen

Es ist nicht notwendig eine ausgefeilte Maussteuerung zu implementieren

- es reicht eine vereinfachte Navigation (z.B. absolut mittels Keyboard) um die Performance testen zu können

Auswahl der Aufgabe

Email an rft@vrvis.at bis spätestens Mo. 7. Dez. 23:59

- Teams von ein bis zwei Studenten (nur in Ausnahmefällen mit erweiterter Aufgabenstellung 3 Studenten)
- für einen Studenten
 - reduzierte Aufgabe z.B. nur 2D Bewegung, Quadtree, oder alles axis-aligned (allgemeine Fälle müssen nicht unbedingt behandelt werden)
- für zwei Studenten
 - allgemeinere Aufgabenstellung, z.B. 3D, Octtree, allgemeine Lage der Geometrie, ausführlichere Analyse (Bericht: 4 Seiten).
- Nachfrage ob eine Aufgabe den Anforderungen entspricht
 - per email oder jeweils nach der Vorlesung

Mündliche Prüfung

- Individuelle Termine bis spätestens 25. März: rft@vrvis.at
- bitte 3 Terminvorschläge bis spätestens zwei Wochen vor dem gewünschten Termin (jeweils Mo-Fr, 10-11:30 oder 14:00-16:30) per email schicken, Ausarbeitung bis 2 Tage vor dem Termin
- Die Prüfung dauert in der Regel 10 Minuten pro Student
- Bei der Abgabe möchte ich das ausgearbeitete Programm laufen sehen: am Studenten-eigenen Notebook oder auf unserem Rechner (Windows 8.1, CPU: i5 4690 @ 3.5 GHz, 16GB RAM, Geforce GTX 970).
- Source-Code muss mit abgegeben werden, ich stelle Fragen dazu
- Weiters noch zwei Fragen zum Stoffgebiet der Vorlesung

Danke für Ihre Aufmerksamkeit!

Robert F. Tobler
rft@vrvis.at

