



## ShadeX Engine Demo

Version 1.0  
7.2.2014

Tristan Bauer  
(E066 932/0426296)

186140 VU Echtzeitgraphik  
Technische Universität Wien  
Institut für Computergraphik und Algorithmen

# 1. Projekt

## 1.1. Bedienung

Die Demo lässt sich im Unterverzeichnis bin über die Datei ShadeXEngineDemo.exe starten. Um den Kernteil der Anwendung zu betrachten ist keine weitere Benutzerinteraktion notwendig, die Kamerafahrt wird automatisiert ausgeführt, und mit Beendigung der automatisierten Kamerafahrt terminiert auch die Applikation. Um zwischen automatisierter Kamerafahrt und einem interaktiven Benutzermodus hin und herzuwechseln ist die Betätigung der Leertaste notwendig. Im interaktiven Modus kann die Kamera mit den Tasten W,A,S,D und der Maus bewegt werden. Es gibt verschiedene Rendermodi, zwischen denen im interaktiven Modus gewechselt werden kann: Taste 1 aktiviert Simulation und Rendering von Rauch ohne Lightshafts, Taste 2 aktiviert Simulation und Rendering von Rauch mit Lightshafts, Taste 3 deaktiviert Simulation und Rendering von Rauch und Lightshafts, Taste 4 deaktiviert Simulation und Rendering von Rauch, und aktiviert Lightshafts. Mit Taste 5 kann die Generierung von neuem Rauch aktiviert/deaktiviert werden. Taste 6 verschiebt das Zentrum der Rauchemission unterhalb der Kamera. Mit Taste 7 wird die Bewegung der Sonne aktiviert/deaktiviert werden.

## 1.2. Die Demo

Aufbauend auf der ShadeX Engine, welche unten näher beschrieben wird, ist eine Demo implementiert worden, welche sich auf einem Terrain mit Grass, Wasser, Rauch und Bergen abspielen wird. Die Dynamik des Rauches wird auf der GPU mittels 3D Texturen realisiert, wobei die Navier-Stokes für newtonsche inkompressible Fluide in Eulerkoordinaten numerisch gelöst werden. Die Darstellung des Rauches erfolgt via Raycasting. Detailreichtum am Terrain in Abhängigkeit von der Entfernung zum Betrachter wird durch Tessellation ermöglicht. Die gesamte Szene wird zu verschiedenen Tageszeiten gerendert, wobei am Skydome Sonnenauf-

und Untergang zu sehen sein werden. Rund um die Sonne sind Light Shafts zu sehen. Detailreichtum der Tiere und Felsen wird mittels Normalmapping realisiert, und durch Vertex Blending werden Tiere animiert. Wasser mit Reflexion und Kaustiken sind zu sehen.

### 1.3. ShadeX Engine

Kernstück des Projekts ist die ShadeX Engine, welche ihrerseits auf dem OpenGL 4.0 core profile aufbaut. Der wesentliche Teil der Engine ist bereits implementiert<sup>1</sup>, und wird voraussichtlich für den Übungsteil der Lehrveranstaltung Echtzeitgraphik keine wesentlichen Erweiterungen erfordern. Das wichtigste Design Ziel bei der Planung dieser Rendering Engine ist es, das Implementieren von Effekten mithilfe von Shader Programmen zu erleichtern. Jeder Effekt ist in einen oder mehrere Pässe und Framebuffer gegliedert, und bietet eine Schnittstelle, welche Rendertargets und Eingabetexturen definiert. Zwischen Pässen, Effekten, Framebuffer und Texturen können Beziehungen festgelegt werden, welche festlegen, welche Eingabetexturen und Rendertargets von den Pässen und Effekten verwendet werden. Jedem Pass können Render Objekte hinzugefügt werden, die Meshes gemeinsam mit optional weiteren Ressourcen kapseln. Ressourcen wie Uniforme Variablen, Texturen und Shader Programme können entweder innerhalb eines Effekts, innerhalb eines Passes, oder innerhalb eines Render Objekts sichtbar sein. Darüber hinaus ist Instanced Rendering möglich, indem einem Render Objekt entsprechende Vertex Buffer zur Verfügung gestellt werden. Das Output eines Passes kann wahlweise in einer Menge von 2D, 3D Texturen, in mehreren Vertex Buffern (via Transform Feedback) abgelegt, oder direkt am Bildschirm dargestellt werden. Darüber hinaus können Meshes im Polygon File Format, Collada (inklusive Bone Hierarchie), und Bilder mit bis zu 16 Bit per Channel geladen und gespeichert werden.

---

<sup>1</sup><http://web.student.tuwien.ac.at/~e0426296/SX%20language.pdf>

## 2. realisierte Effekte

**Rauch** Rauch mit Dynamik, die in Echtzeit auf der GPU berechnet wird [4][2]

**Terrain** Ein Terrain, dessen Detailreichtum sich in Abhängigkeit von der Entfernung zum Betrachter ändert [7]

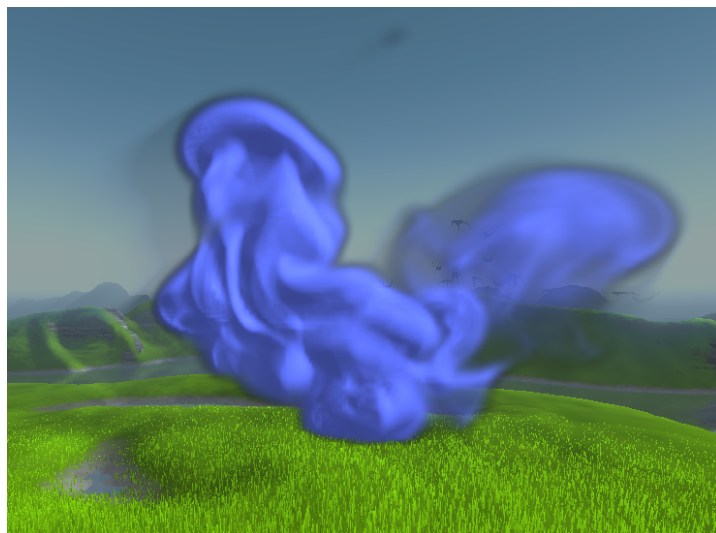
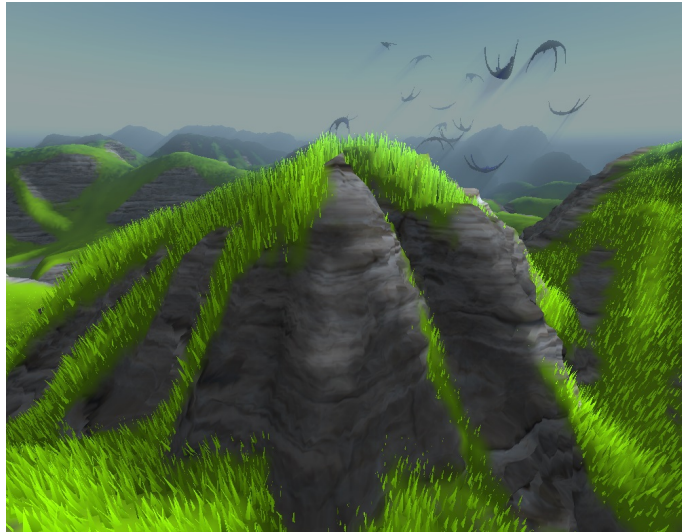
**Wasser** Wasser[5] wird mit Reflexion und Refraktion bei tiefer gelegenen Stellen des Terrains zu sehen sein. Die Animation von Wellen wird durch Verschiebung von Normal Maps implementiert.

**Pflanzen, Tiere** Gräser[3], wie auch Tiere verleihen dem Terrain Leben. Detailreichtum an den Tieren wird durch Normal Mapping, oder durch dynamisch erzeugte Geometrie (Tessellation an den Grashalmen) erzielt. Um der Szene einen lebhaften Eindruck zu verleihen, werden die Pflanzen animiert. Die Bewegung größerer Pflanzen wie auch Tiere wird mit Hilfe von Vertex Blending[1] realisiert.

**Skydome** Ein Skydome[6] mit verschiedenen Tageszeiten wird die Szene umgeben.

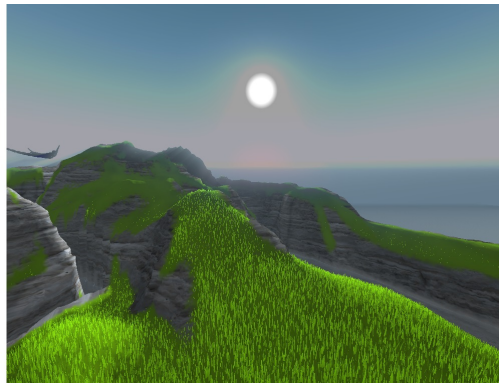
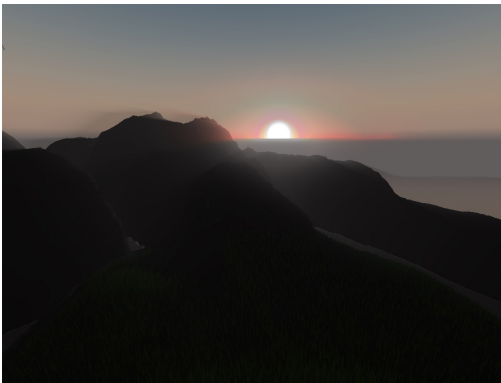
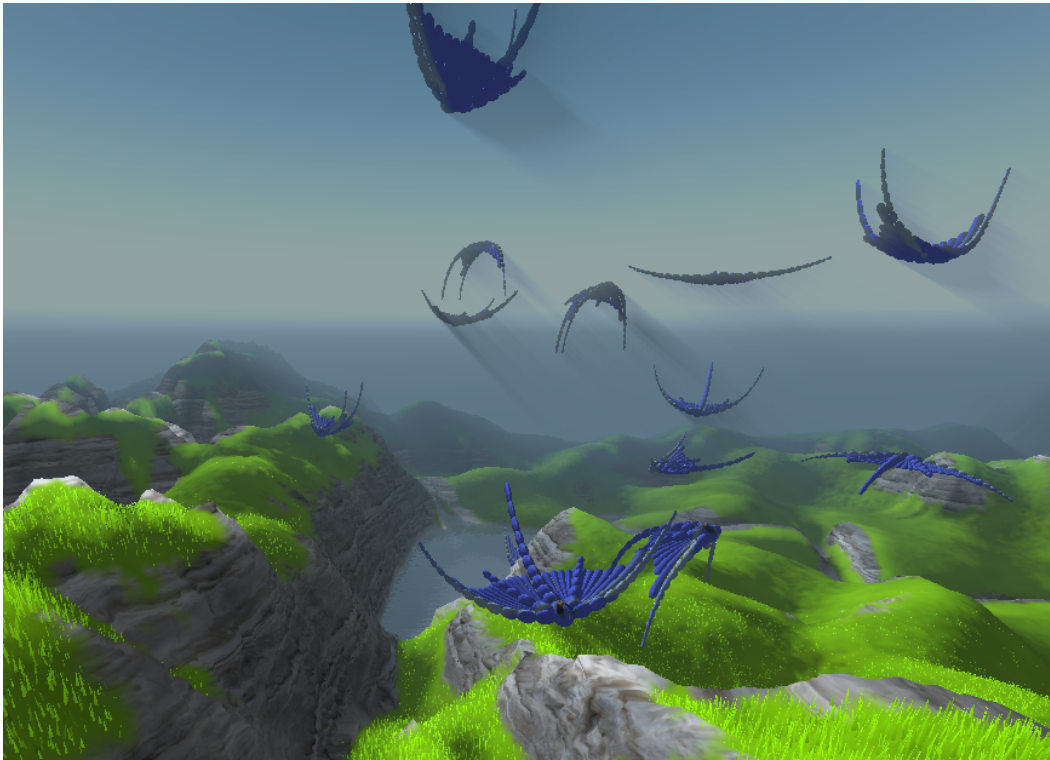
## 2. realisierte Effekte

---



## 2. realisierte Effekte

---



# A. Literaturverzeichnis

- [1] Curtis Beeson. Animation in the "dawn"demo. *GPU Gems*, 2004.
- [2] Robert Bridson and Matthias Mueller-Fischer. Fluid simulation: Siggraph 2007 course notes. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, pages 1–81, New York, NY, USA, 2007. ACM.
- [3] Klemens Jahrmann and Michael Wimmer. Interactive grass rendering using real-time tessellation. In Manuel Oliveira and Vaclav Skala, editors, *WSCG 2013 Full Paper Proceedings*, pages 114–122, June 2013.
- [4] R.W. Johnson. *The Handbook of Fluid Dynamics*. Mechanical engineering. CRC Press, 1998.
- [5] Yann Lombard. Realistic natural effect rendering: Water i. September 2004.
- [6] Alexander Wilkie Lukas Hosek. An analytic model for full spectral sky-dome radiance. *ACM TOG* 31(4).
- [7] Anna Puig-Centelles Miguel Chover Oscar Ripolles, Francisco Ramos. Real-time tessellation of terrain on graphics hardware. *Computers and Geosciences*, September 2011.