

Übungsblatt 10

Ausgabe: 12.01.2011
Einzureichen bis: 19.01.2011, 24:00 Uhr
Besprechung: 21.01.2011 in der Übung

Nutzen Sie zur Strukturierung Ihrer Abgaben Packages, die durch Ihre Namen eindeutig gekennzeichnet sind (Beispiel: `diemkebrunhorn.uebungsblatt1.Aufgabe1a.java`). Exportieren Sie den Quelltext Ihrer Abgabe als ZIP Archiv und senden Sie dieses an `johannes.diemke@informatik.uni-oldenburg.de`.

Aufgabe 32 *GLSL Reflection-Mapping*

Unter dem Reflection-Mapping wird eine effiziente Methode zur Simulation komplexer spiegelnder Oberflächen mittels einer oder mehrerer vorberechneter Texturen verstanden. In dieser Aufgabe sollen Sie die beiden gängigen Varianten, das Spherical Environment Mapping und das Cubic Environment Mapping, als Shader-Programme implementieren.

- a) Implementieren Sie das Spherical Environment Mapping und fügen Sie Ihrer Szene ein Objekt hinzu welches dieses zur Darstellung einer spiegelnden Oberfläche nutzt.
- b) Implementieren Sie das Cubic Environment Mapping und fügen Sie Ihrer Szene ein Objekt hinzu welches dieses zur Darstellung einer spiegelnden Oberfläche nutzt. Verwenden Sie als Cube Map Texturen die Texturen Ihrer Skybox und berücksichtigen Sie, dass sich eine Drehung der Umgebung entsprechend auf die Spiegelung auswirkt.

Aufgabe 33 *GLSL Multitexturing*

In dieser Aufgabe sollen Sie das Multitexturing mittels eines Shader-Programms implementieren. Das Multitexturing bietet die Möglichkeit ein Primitiv gleichzeitig mit mehreren Texturen zu versehen. Eine häufige Anwendung erfährt es bei den beiden als Lightmapping und Detailmapping bekannten Techniken. Zu dessen Nutzung muss eine Grafikkarte allerdings über entsprechend viele Textureinheiten verfügen. Entwickeln Sie ein Programm welches ein beliebiges Objekt mit einer Basistextur (bspw. eine Marmor-Textur) sowie einem Reflection-Mapping versieht. Dadurch erzielen Sie den Effekt, dass sich auf Ihrem texturierten Objekt zusätzlich die Umgebung spiegelt. Verwenden Sie zur Kombination der beiden Texturen das Alpha-Blending und wählen Sie den Parameter α sinnvoll.

Aufgabe 34 *GLSL Bump Mapping*

Das Bumpmapping ist eine Technik zur Darstellung unebener Oberflächen. Dazu werden Normalen-Vektoren in einer so genannten Normalmap abgelegt und zur Lichtberechnung verwendet. In dieser Aufgabe sollen Sie das Bumpmapping-Verfahren mittels der GLSL implementieren. Verwenden Sie dazu im Tangentspace definierte Normalmaps und nutzen Sie das Phong Beleuchtungsmodell für die Lichtberechnung. Es erweist sich als vorteilhaft die Lichtberechnung im Tangentspace durchzuführen. Dazu müssen Sie zunächst im Vertex-Shader die Position des Auges und des Lichtes mittels der inversen TBN-Matrix in den Tangentspace transformieren. Ferner benötigt Ihr verwendeter Vertex-Shader grundsätzlich außer der Normale noch eine Tangente, welche Sie diesem in Form eines Vertex-Attributs übergeben müssen.

- a) Implementieren Sie ein Shader-Programm, welches das zuvor dargestellte Bumpmapping Verfahren umsetzt und texturieren Sie mit diesem eine ebene Fläche.
- b) Texturieren Sie jetzt eine unebene Fläche (bspw. eine Kugel oder einen Zylinder) mit Ihrem Shader-Programm. Dazu müssen Sie je Vertex jeweils einen Normalen- und einen Tangenten-Vektor bestimmen.

Aufgabe 35 *Vorbereitung auf das Einzelprojekt*

In Vorbereitung auf Ihre Einzelprojekte sollen Sie in dieser Aufgabe eine verfeinerte Projektskizze erstellen, die Ihre Projektidee ausführlich beschreibt. Finden Sie dazu zunächst einen Projektnamen und beschreiben Sie Ihre Projektidee. Die Abgabe soll in Form eines PDF-Dokumentes erfolgen.