

OpenGL Aufgabe 4

Andrei Bârsan

November 19, 2012

Aufgabe 10: Licht & Material

a)

Ich habe schon ein Licht mithilfe der Programmable Pipeline implementiert. Es gibt eine "diffuse", eine "ambient" und eine "specular" Komponente. Man kann "L" drücken, um das Licht an- und auszuschalten. Alle Licht-Rechnungen passieren im Shader Programm.

b)

Materialien wurden noch nicht implementiert. Bis jetzt, alle Eigenschaften des Materials stammen von den Licht-Eigenschaften.

c)

Scheinwerferlicht noch nicht implementiert.

Aufgabe 11: Wavefront OBJ Loader

Das System ladet sowohl die Normalen, als auch die Textur-Koordinaten (für die nächsten Aufgaben =).

Aufgabe 12: Vorbereitung auf das Einzelprojekt

Ich werde ein einfaches Computerspiel entwickeln. Es wird ein "Platformer" sein, in einer 3D Welt. Es wird ein Computerspiel sein, aber wenn möglich, möchte ich es auch zu Android portieren. Dazu, muss ich zumindest die folgende Sachen implementieren:

- ein 100% funktionales Renderer das Textur-, Bump- und Normalmapping machen kann (und auch schnell genug um 30 FPS auf einem medium-end Computer zu erreichen)
- Schatten wären auch schön, aber nicht 100% notwendig (ich muss sehen ob es genug Zeit gibt um Schatten zu implementieren gibt)
- Collision detection - entweder selbst implementieren (weil das Spiel ein Platformer ist, wird die Collision-Detection nicht zu schwer sein weil es, theoretisch einfach in 2D passieren wird - mit collision-boxes, und vielleicht ein Quadtree-System um die Collision-Detection ein wenig schneller zu machen), oder ein schon existierendes System wie Bullet integrieren
- Level-Loader - ein System um Spiel-Levels aus (Text-) Dateien zu laden
- Spiel Zustände und Bedingungen - wie bewegt sich man, wie gewinnt man, wie verliert man? Gibt es Punkte, bonuspunkte, combos, etc.?
- Ein Bildschirmverwaltungssystem - wir müssen ein Menuscreen haben, aber auch ein Gameplayscreen, verschiedene Options-screens usw.
- Sehr wichtig - Animationen; ziemlich schwer zu implementieren aber sehr wichtig für ein Computerspiel! Das System muss auch in Stande sein, Animationen zu importieren (es ist ganz besser wenn man die Animationen erst in Maya oder Blender macht, und sie nur danach im Spiel importiert, als wenn man die Animationen hardkodiert)
- GUI Rendering (2D Elemente über ein 3D Szene)
- Sound
- 2D - Kunst - GUI, Logo, etc.