Benutzerdokumentation zum Programmierprojekt „Marching Cubes“

Gruppe 4

Erik Söhnel

Steffen Kersten

Sebastian Wiegand

4. März 2009

Programmbeschreibung

Zuerst wird über *stdin()* eine Liste von Dreiecken eingelesen. Diese werden hiernach auf eine x-y Ebene projiziert und über *stdout()* ausgegeben.

Dies wird über drei Methoden realisiert: *input ()* werden die Dreiecke zeilenweise übergeben und zwar in folgender Struktur: Punkt a, Punkt b, Punkt c. Jeder Punkt wird als drei Leerzeichen getrennte floats dargestellt: *input (3,4 4,6 3,8 1,4 4,5 6,7 3,4 5,6 7,8)*

Die Funktion Funktion *output()* gibt ebenfalls ein Dreieck in oben genannter Struktur zurück, sowie zusätzlich noch die Flächennormale, welche normalisiert und in Form eines Vektors, selbiger besteht aus drei floats, zurück gegeben wird.

Das Programm wird mithilfe eines Scripts, das, je nach Betriebssystem eine Batchdatei (Windows) oder ein Shellscript (Linux) ist.

Zunächst erfolgt eine Ausgabe: "Gruppe 4 Aufgaben 2 - 5. Ließt Dreieckskoordinaten zeilenweise von Stdin ein und transformiert diese in ein Kamerasystem. Gibt projizierte Schirmkoordinaten und Normalenvektor der Dreiecke von hinten nach vorn Zeilenweise auf Stdout aus.
Der Aufruf: gruppe [--help] [Kameraposition] [Blickrichtung]
Optionen: Kameraposition: x,y,z Position der Kamera in der Welt (Vorgabe: Standardposition s)
Blickrichtung: x,y,z Blickrichtung der Kamera (Vorgabe: Standardblickrichtung s)

Hierbei ist zu beachten das die Blickrichtung und die Kameraposition, die jeweils 3 Floatzahlen umfassen, welche durch Kommata separiert sind, **ohne** Leerzeichen zu formatieren sind.
Beispiel: 1,1,1 oder 1.0,-10.98,0.1

Sollten größere Datenmengen anfallen kann mit dem Java-Parameter -Xmx... die Heapgröße gesetzt werden. Anstelle der "..." können beliebige Werte gesetzt werden. K steht für Kilobyte, M für Megabyte, G für Gigabyte. Standardmäßig sind 128 Megabyte gesetzt.

Die benutzte Java-Version ist 1.6.11

Der Verzeichnisbaum

Der Verzeichnisbaum ist wie folgt aufgebaut:

Im Unterverzeichnis "src" befinden sich die Quellen. In "classes" findet man die kompilierten Java-Klassen. In "doc" befindet sich die Javadocs sowie diese Dokument. Unter "samples" sind Beispieldateien vorhanden.
Im Wurzelverzeichnis (das vom Nutzer angelegt werden muss) befindet sich noch ein Build Skript namens build.xml welches mit Ant das Projekt baut.

Link zum Repository

http://github.com/hoeck/psei/tree/stable --funktionierende Version

http://github.com/hoeck/psei/tree/master -- Entwicklungszweig

Kompilieranleitung

Programm:
ins Programmverzeichnis wechseln:
$> ant

Javadoc:
ins Programmverzeichnis wechseln:
$> ant javadoc

Fehlerbehandlung

Zunächst wird eine kurze formale Fehlerbeschreibung ausgegeben und danach der Java-Stacktrace.

"Formatfehler": eine Floatzahl konnte im Input oder als Kommandozeilenargument nicht gelesen werden. Hier sollten die übergeben Parameter geprüft werden.

"Fehler in der Ein/Ausgabe": Hier liegt eine Veränderung des Streams vor, wodurch dieser nicht mehr ordnungsgemäß verarbeitet werden kann.

"OutOfMemoryError": Hier ist die Datenmenge zu groß und passt nicht mehr auf den Heap. daher sollte über den oben beschriebenen Java-Parameter die Heapgröße vergrößert werden.