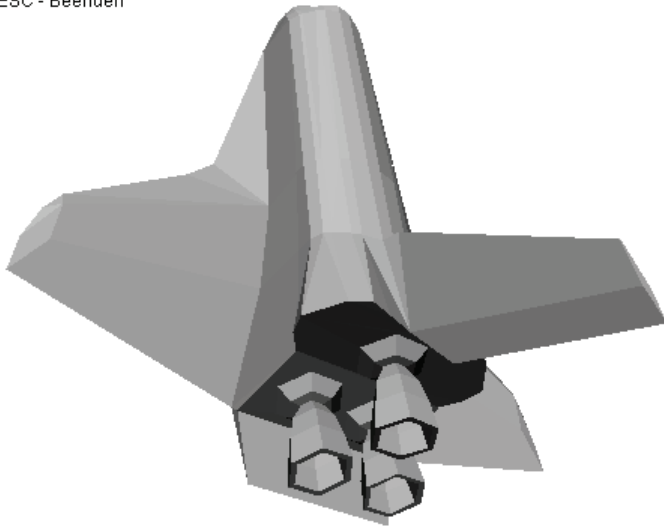


3D-Graphikprogrammierung - Ein Leitprogramm mit Java

r - Alles zuruecksetzen
+/- - Ein- / Auszoomen
n - Eckennr. = false
k - Kanten = false
f - Flaechen = true
x,s - Drehung um x-Achse
y,a - Drehung um y-Achse
c,d - Drehung um z-Achse
ESC - Beenden

Objekt: shuttle.obj
Auge = (0/0/5)



Informationen für die Lehrperson

Worum geht es?

Vom 3D-Kino bis zum 3D-Computerspiel sind virtuelle Realitäten heute weit verbreitet. Die Thematik übt eine grosse Faszination auf Jugendliche aus, weil sie grundlegende Fragen nach der Funktionsweise des Gesichtssinns und der dabei auftretenden perspektivischen Abbildung der umgebenden Welt mit der Möglichkeit verbindet, die eigene Kreativität bei der Entwicklung einer virtuellen Welt einzubringen. Dieses Leitprogramm konzentriert sich auf die dafür notwendigen Grundlagen seitens der Informatik und der Mathematik. Dies setzt ein Verständnis der perspektivischen Abbildung und deren Umsetzung mit der objektorientierten Programmiersprache Java voraus. Ausserdem werden einfache geometrische Abbildungen wie räumlichen Streckungen und Drehungen modelliert und in Java implementiert. Schliesslich wird die Sichtbarkeit der Oberflächenpolygone von dreidimensionalen Polyedern untersucht. Dabei erhalten die Schülerinnen und Schüler einen Einblick in die Graphikprogrammierung im dreidimensionalen Raum. Im Anschluss an das Leitprogramm empfiehlt sich die Verwendung einer professionellen 3D-API wie beispielsweise Java 3D oder eines Raytracers wie POV-Ray.

Voraussetzungen

Das Leitprogramm arbeitet mit der von Aegidius Plüss entwickelten GPanel-Graphik¹, welche einen direkten Zugang zu hochwertigen Graphikbefehlen in einem frei wählbaren Koordinatensystem ermöglicht. Diese Hilfsklassen sind auf der Homepage von Plüss mit javaDoc ausführlich dokumentiert. Für die Speicherung der Oberflächenpolygone wird im 4. Kapitel des Leitprogramms ein TreeSet eingesetzt. Dieser Datentyp setzt voraus, dass die Schülerinnen und Schüler mit einem Iterator umgehen können. Ausserdem werden von der Mathematik her die Grundlagen der Vektorgeometrie vorausgesetzt.

Lernziele

- Die SuS verstehen die technischen Voraussetzungen für virtuelle Realitäten.
- Die SuS können einfache dreidimensionale Objekte als Vektorobjekte digitalisieren und mittels einer Zentralprojektion visualisieren.
- Die SuS kennen die Rückseitenentfernung und ein Beispiel eines Prioritätsalgorithmus zur Lösung des Sichtbarkeitsproblems und können diese Verfahren in eigenen Worten erklären.

Kompetenzen

- Die SuS erwerben sich anhand des Leitprogramms grundlegende Kenntnisse über Modellierung und Implementierung einer geometrischen Problemstellung, so dass sie ein Polyeder digitalisieren und mithilfe abstrakter Datentypen in Java modellieren und visualisieren können.
- Die SuS trainieren die objektorientierte Arbeitsweise mit vordefinierten Klassenbibliotheken (z.B. für Punkte, Vektoren und Matrizen) und der zugehörigen javaDoc, so dass sie externe Klassen für ihre Programmieraufgaben gewinnbringend einsetzen können.

Weiterführende Literatur

- Frank Klawonn. Grundkurs Computergrafik mit Java. Vieweg+Teubner Verlag, Wiesbaden 2009. Homepage: <http://public.rz.fh-wolfenbuettel.de/~klawonn/computergrafik/>
- Marius Apetri. 3D-Grafik Programmierung. mitp, Heidelberg 2008.
- Wolf-Dieter Klix. Konstruktive Geometrie: darstellend und analytisch. Fachbuchverlag Leipzig, München 2001.
- Aegidius Plüss. Java exemplarisch. Oldenbourg 2004. Homepage: <http://www.aplu.ch>.
- Spezifikation des Wavefront Object Formats:
<http://local.wasp.uwa.edu.au/~pbourke/dataformats/obj/>.

¹ Die GPanel-Graphik ist Teil der Hilfsklassen zum Buch „Java exemplarisch“ von Aegidius Plüss.

3D-Programmierung: Stand der Dinge im Sommer 2009

Auf SwissEduc gibt es bereits ein sehr umfangreiches Leitprogramm zur Einführung ins Programmieren mit dem professionellen Raytracer POV-Ray¹. Dabei geht es darum, den Lernenden mit POV-Ray Grundkonzepte des Programmierens wie die Sequenz, die Selektion und die Iteration anschaulich zu vermitteln. Die Funktionsweise des Raytracers und die zugrundeliegenden mathematischen Modelle werden dabei bewusst ausgeklammert. Daneben gibt es zahlreiche Websites zur Graphikprogrammierung mittels APIs wie OpenGL oder Java3D, mit deren Hilfe Softwareentwickler schrittweise den Umgang mit diesen mächtigen Tools erlernen können. Im Folgenden ein paar Beispiele (Stand 31.08.09):

- <http://www.opengl.org/>: Homepage zur Graphik-API OpenGL
- <https://java3d.dev.java.net/>: Projekthomepage von Java 3D
- <http://www.jmonkeyengine.com/>: Homepage einer Open Source 3D-Spiel-Engine

Hinweise zur Bearbeitung des Leitprogramms

Für dieses Leitprogramm wurden Hilfsklassen entwickelt, mit denen beispielsweise Koordinatenwerte aus einem Datenfile im Wavefront Object Format eingelesen werden können. Diese Klassen erlauben es den Lernenden, sich auf die geometrische Modellierung zu konzentrieren. Wie in der Computergraphik üblich, kann auf die Komponenten der Vektoren nicht nur mittels getter- und setter-Methoden sondern auch direkt über die entsprechenden Instanzvariablen zugegriffen werden, beispielsweise `v.x` statt `v.getX()`. Dies ermöglicht eine einfachere, kürzere Darstellung der geometrischen Transformationen in Anlehnung an die Vektorschreibweise. Zu den Hilfsklassen existiert eine Dokumentation, welche mit javaDoc erstellt wurde. Weitere Informationen dazu sind im Anhang A des Leitprogramms zu finden. Sämtliche Lösungen zu den Programmieraufgaben liegen auch in digitaler Form vor. Die Dateinamen der Beispiellösungen sind von der Form `Polyeder3D_vX.java`.

Übersicht

1. Kapitel: Perspektivische Abbildung von Polyedern (2 – 3 Lektionen)

Im 1. Kapitel geht es darum, die perspektivische Abbildung eines Würfels in einer vorbereiteten Java-Umgebung zu implementieren. Die Umgebung klammert Fragen der GUI-Programmierung aus, so dass sich die SuS auf die 3D-Graphikprogrammierung konzentrieren können. Es sollen Datenstrukturen zur Repräsentation von Polyedern entwickelt werden. Wie weit die SuS auf diesem Weg selbst gehen sollen, ist individuell steuerbar, da sie nach erfolgreicher Bearbeitung der Grundaufgabe jederzeit zum nächsten Kapitel wechseln können.

¹ Fabian Hauser, Claudia Käppeli. Mit Povray Grafik programmieren: Ein Leitprogramm. Siehe <http://www.swisseduc.ch/informatik/programmiersprachen/povray/>.

2. Kapitel: Ein Dateiformat für 3D-Objekte (2 Lektionen)

Im Zentrum des 2. Kapitels steht das so genannte Wavefront Object Format, ein Dateiformat für 3D-Vektordaten. Die Daten werden dabei im ASCII-Format gespeichert und können jederzeit mithilfe eines Texteditors modifiziert werden. Da das Format weit verbreitet ist, können auch professionelle Graphikprogramme wie beispielsweise Blender¹ zum Erstellen und Bearbeiten der Dateien eingesetzt werden.

3. Kapitel: Geometrische Transformationen (2 – 3 Lektionen)

Im 3. Kapitel stehen räumliche Streckungen und Drehungen der Polyeder auf dem Programm. Der Einfachheit halber werden die Objekte am Nullpunkt gestreckt, bzw. um die Koordinatenachsen gedreht. Nach einer kurzen Einführung in die Matrizenrechnung werden die SuS angeleitet, eine Hilfsklasse Matrix zu benutzen, um die Streckungen und Drehungen in Java zu implementieren. Darüber hinaus kann mit weiteren linearen Abbildungen experimentiert werden, wobei die Auswirkungen im Projektionsprogramm sofort sichtbar werden. Das 3. Kapitel ist daher optimal geeignet für eine fachübergreifende Zusammenarbeit mit der Mathematiklehrperson (Matrizenrechnung).

4. Kapitel: Darstellung von Oberflächenpolygonen (2 – 3 Lektionen)

Das 4. Kapitel schliesst den obligatorischen Teil mit der Darstellung von Oberflächenpolygonen ab. Dabei wird das Sichtbarkeitsproblem für konvexe Polyeder mithilfe der Rückseitenentfernung gelöst. Dazu sollen die SuS mit dem Vektorprodukt die Normalenvektoren der Oberflächenpolygone berechnen. Für den Fall, dass das Vektorprodukt nicht bekannt ist, können die Normalenvektoren auch mit der Methode `getNormal(Face)` der Hilfsklasse Polyeder geladen werden. Dafür müssen sie allerdings vorgängig mittels `generateNormals()` berechnet werden. Entsprechende Lösungen finden Sie in den Dateien `Polyeder3D_v07b.java` und `Polyeder3D_v08b.java`. Ausserdem wird ein Prioritätsalgorithmus zur Lösung des Sichtbarkeitsproblems für Polyeder mit Ein- oder Ausbuchungen vorgestellt.

5. Kapitel: Oberflächenschattierung – ein Ausblick (2 Lektionen)

Das 5. Kapitel ist als Additum für schnelle Schülerinnen und Schüler gedacht. Hier geht es um das sogenannte Shading, d.h. um die Schattierung der Oberflächenpolygone. Dabei wird exemplarisch das Verfahren der konstanten Schattierung² mithilfe der Normalenvektoren der Oberflächenpolygone behandelt.

¹ Vgl. <http://www.blender.org/>.

² engl. flat shading