Rota – Abgabe 2

Computergraphik 2 LU

20.06.2008
Georg Zankl
Timo Kropp

Kurzbeschreibung

In der aktuellen Version von Rota spielt man einen Reifen aus der 3rd Person Perspektive. Der Spieler kann diesen frei bewegen und dabei Gegenstände(eine Kugel und ein Fass) verschieben.

Ziel des Spiels ist es die Gegenstände auf die farblich Markierten Ziele zu schieben. Wenn dies geschafft ist Die Schwierigkeit dabei besteht darin, dass diese Ziele irgendwo im Raum (vielleicht sogar an der Decke) sind. Es gibt Rampen über die man die Wände entlang fahren kann. Gegenstände allerdings kommen nicht über diese rampen, Gegenstände muss man an eine Wand schieben und dann, wenn man selbst an dieser entlang fährt, den Gegenstand von dieser Wand entfernen. Der Gegenstand ist daraufhin auf die Wand ausgerichtet und kann verschoben werden, als wäre die Wand nun der eigentliche Boden.

Steuerung

Der Reifen wird mit W nach vorne und S nach hinten bewegt. Mit S dreht man es nach links und mit D nach rechts. Mit der Maus kann man die Kamera in alle Richtungen frei um den Reifen rotieren. Sobald der der Spieler sich mittels W nach vorne bewegt, schwenkt die Kamera in die Richtung der Bewegung.

Tastenbelegung der verschiedenen Modi und informationen:
F2 Framerateanzeige an/aus
F3 Wire Frame Modus an/aus
F4 Textuqualität Nearest Neighbor/Bilinear
F5 MipMapping Qualität Aus/Nearest Neighbor/Linear
F9 Transparenz an/aus
F10 Collision Meshes/Spheres an/aus

Kurzbeschreibung der Implementierung

Das Kameramodell wurde mit Hilfe von gluLookAt realisiert, dafür wird View- und Up-Vektor gespeichert und mit einer Rotationsmatrix multipliziert, wenn eine Rotation angefordert wird. Um eine flüssigere Kamerabewegung zu erreichen wurden Klassen für interpolierte Vektoren erstellt, bei denen mit Hilfe von Arrays eine Cosinus Interpolation durchgeführt wird.

Für die Objekte gibt es eine Klasse GeMesh, welche sowohl Listen handhabt, als auch die Collision Detection und Response übernimmt (was in dem Obertyp MBounding geschieht).

Die Collision Detection funktioniert bisher zwischen 2 Kugeln und zwischen einer Kugel und einem Dreieck/Mesh. An den Kanten von Dreiecken gibt es noch kleine Probleme den richtigen Bewegungsvektor auszurechnen, daher wurde dieser Teil auskommentiert. Die Collision Response arbeitet bisher rein mit Geschwindigkeiten (noch keiner Kraft/ Beschleunigung). Bei einem GeMesh bzw. MBounding Objekt werden auch Flags übergeben, welche festlegen, ob ein Objekt durch andere Objekte verschiebbar ist, ob sich der Bewegungsvektor bei einer Rotation ebenfalls ändern soll oder ob ein bewegliches Objekt gegenüber anderer beweglichen Objekte dominant sein soll, d.h. ob andere Objekte bei einer Kollision einfach weggeschoben werden sollen.

Sowohl Texturen als auch Materialien werden von der Klasse GeMesh behandelt, dabei gibt es separate Klassen, die man an setter Methoden des Meshs übergibt.

Die Steuerung wurde mit Hilfe von Polling in einer „Input“ Klasse realisiert, welche die Callback Funktionen mit GLUT setzt und die Kamerarotation, sowie die Bewegung des Reifens. Der Mittelpunkt der Kamera wurde einfach über einen Pointer an die Position des Reifens fixiert.

Tools

Modelliert wurde mit 3D Studio Max 9. Die Modelle wurden in das FBX Format exportiert und mit hilfe des FBX SDKs (http://www.autodesk.com) importiert. Texturen wurden mit dem targa Image loader aus dem FBX SDK Beispiel ViewScene geladen.