TubeRacer

Spieldesign

Leider war unsere Steuerung (wie von euch angekuendigt) nicht brauchbar. Deshalb haben wir vor kurzer Zeit damit begonnen, eine neue Steuerung zu implementieren. Leider sind wir damit mit nicht ganz fertig geworden...

Constrains...

Der Player befindet sich auf einem Pfad durch die Tube. Er sich um den Mittelpunkt drehen und sich von ihm entfernen, also an den Rand zufliegen. Der Player ist stets nach der Tube ausgerichtet.

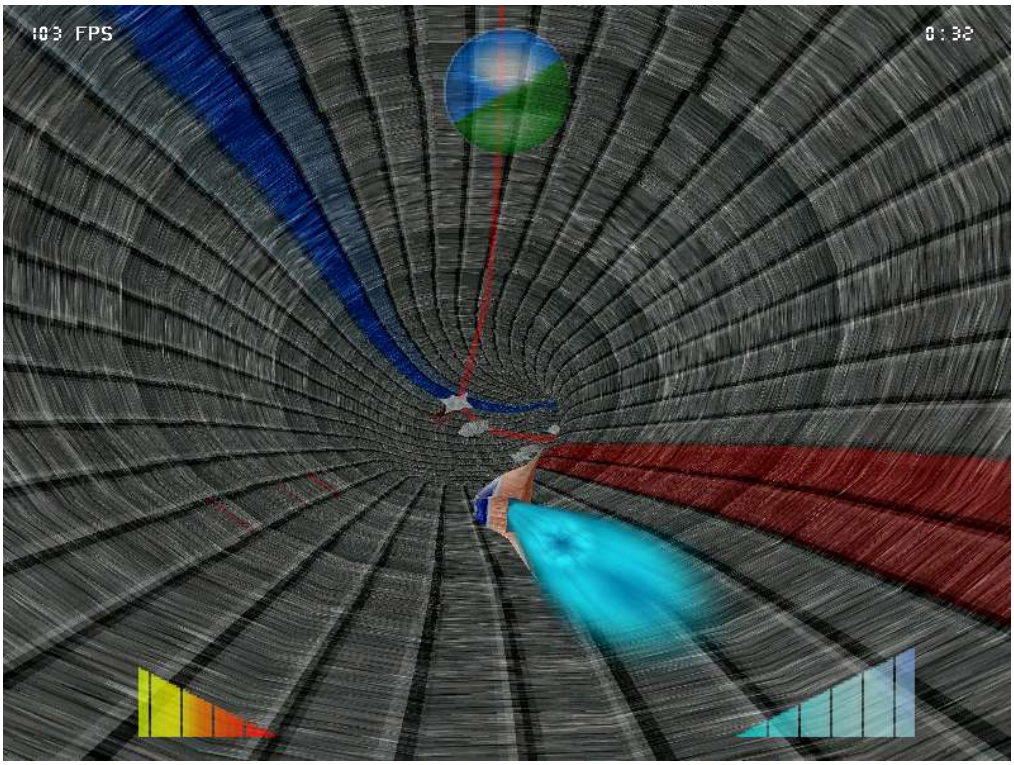

Bild 1

Was auf Bild 1 zu sehen ist:

CG 2/3 Übungen, SS 2004
Abschnitt Nr. 3
Pinter, Bernd, 881, 9725644, bernd@a-theory.tuwien.ac.at
Prosenc, Robert, 534, 9725172, robert@a-theory.tuwien.ac.at

Project-titel: TubeRacer

TubeRacer

Die Tube
Sie bildet den Rahmen, die Welt fuer das Spiel.

Der Pfad
Die Mitte der Tube ist durch die rote Linie dargestellt.

Der Player
Bewegt sich parallel zum Pfad. Er kann die Entfernung zum Pfad (UP, DOWN) aendern und sich um diesen drehen (LEFT, RIGHT).

Die Enemies
Weiter vorne sieht man ein paar Enemies - manche verbleiben an einem Ort, manche bewegen sich zyklisch um einen Punkt, und manche fliegen direkt auf den Player zu.

Das HUD
Links unten wird die Energie des Players dargestellt - bei jeder Kollision mit einem Enemy verliert der Player Energie. Ist diese verbraucht, so explodiert der Player. Rechts unten wird die Geschwindigkeit des Players angezeigt. Beide Anzeigen verwenden Transparenzen.
Oben links wird die Frame-Rate angezeigt. Bei dem Screenshot oben sind das zB: 103 fps (!).
Oben rechts ist die Zeit eingeblendet.
Oben in der Mitte ist die Neigungsanzeige.

Nichttriviale Objekte
Alle verwendeten 3D-Models wurden in Maya erstellt. Texturen aus Photoshop.
Die Models werden als Lightwave obj-file eingelesen. Die Klasse dazu (lesen, schreiben, render) ist selbst geschrieben.

Beschleunigung der Sichtbarkeitsberechnung
Spiel-optimierte Variante von View-Frustum-Culling ist implementiert (laesst sich mit FXX an/ausschalten): es wird nur jener Teil der Tube gerendert, der sich unmittelbar vor dem Player befindet.
Im Wireframe-Modus ist das Culling gut sichtbar.

CG 2/3 Übungen, SS 2004
Abschnitt Nr. 3
Pinter, Bernd, 881, 9725644, bernd@a-theory.tuwien.ac.at
Prosenc, Robert, 534, 9725172, robert@a-theory.tuwien.ac.at

Project-titel: TubeRacer

TubeRacer

Transparenz-Effekte
HUD ist transparent, genauso wie die jeweilligen Partikel-Systeme sowie die Teilabschnittsmarker der Tube ('10%', ' 20%',...)

Experimente mit OpenGL
Vertex Arrays vs. Immediate Mode Triangles - umschalten mit FXX (nur die Models, nicht die Tube)
Mip Mapping - umschalten mit FXX
Display Lists - kann man nicht abdrehen.... die ASCII-Character werden als Display Lists verwaltet.
siehe Red Book

Tastaturbelegung:
F1 - Hilfe (falls vorhanden),
F2 - Framerateanzeige ein/aus,
F3 - Wire Frame ein/aus,
F5 - Mip Mapping ein/aus,
F6 - Vertex Arrays vs. Immediate Mode,
F8 - View frustum culling ein/aus,
F9 - Transparenz ein/aus,
F12 - Pfad anzeigen,
UP - Player richtung Pfad bewegen
DOWN - Player weg vom Pfad (zurm Tuberand hin) bewegen
LEFT - Rotation nach links
RIGHT - Rotation nach rechts
SPACE - Beschleunigen
Y (oder Z) - Schiessen
P - Power Mode
K - Player suicide

Ini - File
Das Programm liest seine Parameter von der Textdatei tuberacer.ini aus. Fuer ein schoenes Menue fehlte leider wieder einmal die Zeit. Der Aufbau des Files ist denkbar einfach:

1..9	level
int	tubesegment_count
1 | 2	window/fullscren

Level 1 ist eine Gerade, Level 9 ist sehr kurvig. Es gilt: Je hoeher der Level, desto hoeher sollte der tubesegment_count sein.

CG 2/3 Übungen, SS 2004
Abschnitt Nr. 3
Pinter, Bernd, 881, 9725644, bernd@a-theory.tuwien.ac.at
Prosenc, Robert, 534, 9725172, robert@a-theory.tuwien.ac.at

Project-titel: TubeRacer

TubeRacer

Spezialeffekte

Transparenz

Die Transparenz der Polygone wird verwendet, um Partikel und Anzeigen dynamisch auszublenden. Transparenz auf den Texturen (32bit-TGA) fuer Billboards und das HUD.

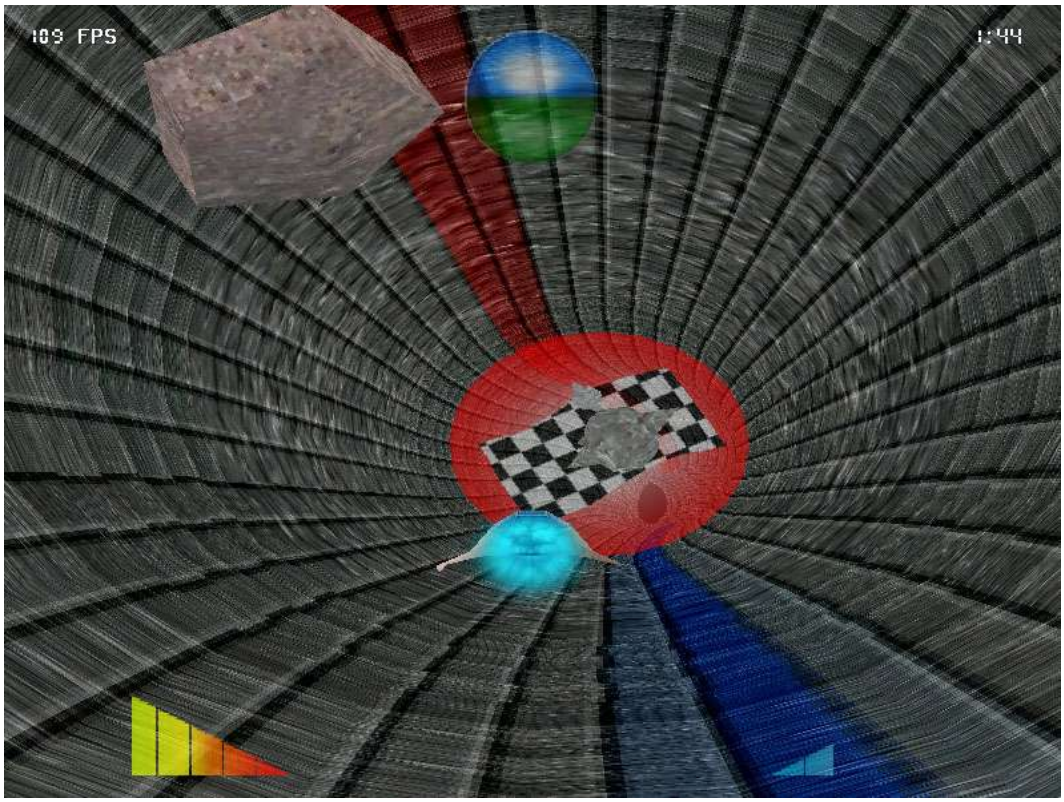

Bild 2: Transparenz

Explosion

Die Gegner explodieren, wenn sie abgeschossen werden (oder wenn sie den Player zu nahe kommen, im PowerMode). Natuerlich explodieren sie auch, wenn sie mit dem Player kollidieren.
Die Exlposion selber wird auf Billboards eingeblendet. Die Texturen auf den Billboards werden animiert (ein Texturfile beinhaltet 9 bzw. 16 frames). Die explosion erzeugt mehrere Billboards, die entweder einen Feuerball oder eine Rauchwolke anieren und sich dabei vom Explosionspunkt entfernen, dabei werden sie transparenter. Der Zufall spielt eine entscheidende Rolle.

CG 2/3 Übungen, SS 2004
Abschnitt Nr. 3
Pinter, Bernd, 881, 9725644, bernd@a-theory.tuwien.ac.at
Prosenc, Robert, 534, 9725172, robert@a-theory.tuwien.ac.at

Project-titel: TubeRacer

TubeRacer

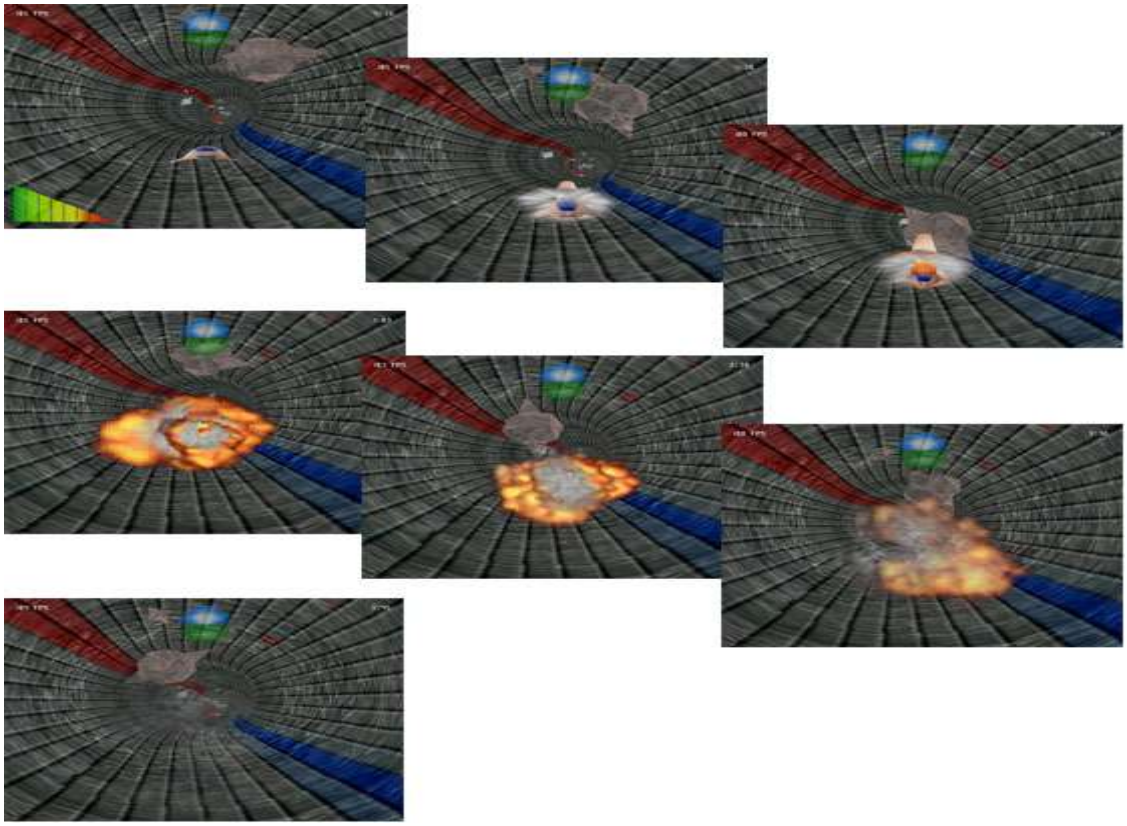

Bild 3: Explosion des Players

Rauchspuren
Vapor-Tails wurden nach dem Paper "Advanced Graphics Programming Techniques Using OpenGL" Kapitel 11 geloest. Partikel werden erzeugt. Zu Beginn sind sie opaque und klein. Im Laufe der Zeit werden sie Transparenter und groesser. Die Partikel sind wieder Billboards mit einer 32bit-TGA Textur. Siehe Bild 1.

Komplexe Partikelsysteme
Siehe Explosion. Mehere Typen von Partikel (Rauch, Feuer) werden mit zufaelliger Groesse, Lebensdauer, Wachstumsfaktor und Initialgeschwindigkeit erzeugt.

Test der Effekte
PowerMode [p] einschalten. Gas geben [SPACE], lenken und schiessen [y|z]

CG 2/3 Übungen, SS 2004
Abschnitt Nr. 3
Pinter, Bernd, 881, 9725644, bernd@a-theory.tuwien.ac.at
Prosenc, Robert, 534, 9725172, robert@a-theory.tuwien.ac.at

Project-titel: TubeRacer

TubeRacer

Zusatztools

GLUT
fmod

CG 2/3 Übungen, SS 2004
Abschnitt Nr. 3
Pinter, Bernd, 881, 9725644, bernd@a-theory.tuwien.ac.at
Prosenc, Robert, 534, 9725172, robert@a-theory.tuwien.ac.at

Project-titel: TubeRacer