Beschreibung des Spiels

Compilieren

Leider sind wir (wegen MMI, VirtualReality bei Prof. Schmalstieg. wir erweitern gerade die studiertube um ein neues Menü-System) gezwungen, anstelle vom geforderten *VisualStudio .NET*, das alte *Visual Studio 6.0* zu verwenden. Auch war der Übungsrechner bis zur Woche 18 nicht in betrieb, weshalb wir die kompatibilität der beiden Versionen nicht testen konnten :-(
Dennoch sollte eine Import des VS 6.0 Projekts in VS .NET kein Problem sein - angeblich. Wir hoffen...

Des weiteren gehen wir davon aus, dass auf der Zielmaschine eine glut.h im GL-Include-Verzeichnis des Compilers liegt (also <GL/glut.h>).

Wir haben ausserdem das VC-Projekt so angelegt, dass eine Konsolenanwendung (ja, das Teil haeißt so bei MS?!) erzeugt wir. Glut startet auch noch nicht im FullScreen-Modus. Als Gegenleistung erhalten wir aber eben eine Konsole, auf der wir Debug-Outs schreiben (sind fast alle für diese Abgabe abgeschaltet). Wir hoffen, dass sie bei dieser Abgabe 2 kein Problem darstellt. Für die Abgabe 3 wird das VC-Projekt selbstverständlich dahingehend geändert, dass Glut das OpenGL-Fenster im FullScreen (mit Game-Extention) startet.

Das Compilat

Das Verzeichnis bin/ beinhaltet sämtliche libraries, textueren, meshes, und natuerlich auch das ausführbare Programme (tuberacer.exe) selbst.
Das Programm wurde als „Release“ im VC erstellt. Die Debug-DLLs sollte daher nicht nötig sein.

Steuerung

Cursor- Links/Rechts lenkt das Schiff nach links bzw. rechts.
Cursor- Up/Down lenkt das Schiff nach Unten bzw. Oben.
Space gibt Gas.

Wenn das Schiff einlenkt, macht die Kamera diese Bewegung mit. Tatsächlich „verfolgt“ die Kamera den Spieler. Sie tut dies allerdings Verzögert. So sieht der Spieler auch, dass sich sein Schiff in die Kurve legt.
Derzeit haben wir 2 verschiedene Modis, wie die Kamera das Schiff verfolgt. Wir sind da noch am experimentieren, was am Schluß wirklich gut aussieht.

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

Project-titel: TubeRacer

Beschreibung des Spiels

F9 erster Verfolger-Modi der Kamera. Dieser ist vom Start des Programms an aktiv.
F10 zweiter Verfolger-Modi der Kamera. Sieht irgendwie netter aus....

F1 zeigt die frames per second auf stdout.

3D-Models

Alle 3D-Modelle wurden in Maja5 als Polygone erstellt. Anschließend wurden die Modelle triangluliert und simplifiziert (nur Dreiecke als Primitive und Polycount gering halten!).
Als guter letzt wurden die Modelle als *Lightwave OBJ Files* exportiert. Diese werden von unseem (selbgeschriebenen!) Loader gelesen. Dieser Loader stellt die Polygone derzeit nur direkt in GL. Also noch keine Vertex/Normal/TextCoord Arrays!

Texturen

Texturen wurden von Maja gemappt (Autotexture: funkt sehr gut!). Das Mapping der Polygone auf die Textur wurde als tiff exportiert und mit Photoshop befüllt. Texturen funktionieren schon (ohne Mipmapping). Das Schiff hat schon seine Textur im Spiel.

weitere Features

Tube. Die Spielwelt.

Die Tube besteht aus einzelnen kurzen Segmenten. Zur besseren Sichtbarkeit wird (derzeit) nicht das gesamte Segment gerendernt, sondern nur 2/3 davon. Es wird jedes 3. Polygon invertiert gezeichnet. So sieht man auch die Tube, wenn man mal rausfliegt. Dies ist derzeit noch möglich, da es keine CollisionDetection gibt!

Die Tube wird in jedem Spiel zufällig erzeugt. Dafür gibt es 9 Schwirigkeitgrade.
Level 1: eine Gerade
Level 9: eine einzige, wilde Kurve in allen 3 Richtungen (x,y,z)
Für diese Abgabe wird eine Tube mit Level 1 erzeugt. Dies kann im File CGameEngine.cpp geändert werden. Der entsprechende Aufruf ist in der Methode
void CGameEngine::init3DObjects () zu finden.

Die Tube kann sich des weiteren selbst abspeichern und wieder laden. Ausserdem (eh nur für Debug-Zwecke, aber trotzdem nett) kann die Tube sich selbst als Lightwave OBJ File exportieren. Dann kann das Teil auch im Maja praktisch studiert werden!

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

Project-titel: TubeRacer

Beschreibung des Spiels

Beleuchtung:

Es sind permanent 3 Lichter im Einsatz. Dabei werden alle 5 Segmente ein Licht im Zentrum platziert. Damit dies nachvollziehbar ist, wird (nur in dieser Abgabe!) bei jedem Licht eine kleine Kugel (glutSolidSphere(1,10,10)) platziert.
Wenn das Schiff ein Licht passiert, so wird dieses Licht neu positioniert. und zwar hinter dem Licht, dass noch am weitesten weg war. Die Lichter reisen also mit dem Schiff mit.

Texturen:

Die Texturen werden alle beim Start der Applikation als GL-Texture-Objects geladen und von einer zentralen Klasse verwaltet (CtextureObjects). Jedes 3D-Objekt, dass eine Texture braucht, fragt diesen Container nach seinem GL-Texture-Index.

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

Project-titel: TubeRacer