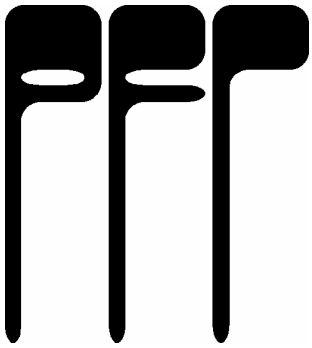

Abenteuer im sehr naiven Kosmos

Genre:	Geschicklichkeit
Ziel:	Sachen Sammeln

Axel Goldmann, e9607701@student.tuwien.ac.at

1 Realisierte Punkte aus der Angabe

1.1 Gameplay

Ziel des Spiels ist es, alle auf Planet und Mond befindlichen Canister einzusammeln. Das Spiel ist verloren, wenn der Treibstoff aufgebraucht ist, zu hart das Fahrzeug gelandet wurde, oder im positiven Fall, alle Cansiter eingesammelt wurden sind. Das erfordert natürlich eine Landung am zweiten Himmelslörper, dem kleineren (grauen) Mond. Eine art Zentralgestirn, an einen Pfirsich erinnerd, ist auch da, weder schädlich noch nützlich.

Es gibt drei Kameraansichten (Steuerung siehe unten):

- Von oben über das Fahrzeug auf den jeweils nächstgelegenen Himmelskörper
- Nach vorne (ähnlich einem Blich aus der Kanzel) um das Navigieren und ausrichten auf der Reise zwischen Mond und Planet zu erleichtern
- Außerdem kann jederzeit frei herumgeblickt werden.

Um in der Nacht besser zurechtzukommen, kann nun eine nach unten gerichtete Lande-/Suchbeleuchtung eingeschalten werden (braucht aber wie im richtigen Leben auch etwas Energie).

Die Steuerung ist jetz mit der Maus sehr viel leichter und auch spaßiger, man kann sie aber durch ungüstige Handhabung invertieren, das ist nichtmehr behoben worden.

1.2 Nichttriviale Objekte

Einziges komplexeres Objekt ist das modellierte Raumfahrzeug.

1.3 Animierte Objekte

Sind nicht vorhanden, es sei denn, man betrachtet das gelandete Raumfahzeug als solches, da es dann lokal zum Himmelskörper behandelt wird.

1.4 Beschleunigung der Sichtbarkeitsberechnung

Nicht implementiert. Der Octree um die Himmelskörper dient nur der Collisiondetection.

1.5 Transparenz-Effekte

Abgaspartikel sowie Textnachrichten sind transparent gemacht.

1.6 Experimentieren mit OpenGL

Gezeichnet wird hauptsächlich mit Array Lists. VBOs sind nicht enthalten. Für das Shadow Mapping wurde ein FBO eingesetzt.

1.7 Spezialeffekte

1.7.1 Shadow Mapping

Es war geplant von überall nach überallhin Schatten zu werfen. Das war aber zuviel. Ich hab mich darauf beschränkt, das Raumfahrzeug und die Canister auf die Himmelskörper Schatten werfen zu lassen. Gerade bei Landungen war das ein Effekt, der sehr abging. Es hat versuche gegeben, das Gelände selbst Schatten werfen zu lassen. Das hab ich aber wegen den Artefakten bei spitzen Winkeln nicht hingekriegt. Es wären dadurch interessantere Geländestrukturen möglich gewesen (Krater und Schluchten statt dem Noise).

Zur Realisierung wurde ein Shader eingesetzt. Der eine Shader nimmt dabei auch alle anderen Aufgaben auf sich (Lighting, Texture Mapping).

Der Tiefenbuffer ist an ein FBO angeknüpft. Die Lichtkamera im ersten Pass ist immer auf das Raumfahrzeug gerichtet. Deshalb verlieren nahegelegene Canister manchmal ihren Schatten. Damit das Hinausragen aus der Lichtkegel keine langen Artefakte ergibt (Clamping), wurde die Tiefentextur mit einem Rand versehen.

Der Bias, der dafür sorg, daß die Schatten nicht mit den Oberflächen un Konflikt kommen ist fix eingestellt.

1.7.2 Partikel

Die Abgasstrahlen sind die einzigen temporär auftretenden Objekte und werden mit einem kleinen Geschwidigkeitsvektor aus der Düse gelassen. Ihre Farbe ist abhängig von ihrer verbleibenden Lebensdauer.

1.7.3 Per Fragment Lighting

Zusätzlich zum Shadow Mapping ist im Shader noch das Per Fragment Lighting enthalten. Leider paßt was nicht mit dem spekularen Anteil.

2 Verwendetes

Externe Teile sind:

- lodepng: ein PNG Bild Lader auf Quellcodeebene
  http://members.gamedev.net/lode/projects/LodePNG/

- rply: ein PLY Objektlader auf Quellcodeebene
  http://www.tecgraf.puc-rio.br/~diego/professional/rply/

- glut: openGL Komfortpacket, hier statisch gelinkt (glut32.lib); von GLEW inkludiert.
  http://www.xmission.com/~nate/glut.html

- glew: The OpenGL Extension Wrangler Library (dynamisch in glew32.dll)
  http://glew.sourceforge.net/

Modelliert wurde der Canister und das Raumfahrzeug mit Blender (www.blender.org),

3 Steuerung

ESC	raus
SHIFT-F	Fullscreen an/aus
SHIFT-C	ungebundene Camera, Mouse look (siehe Steuerung unten)
SHIFT-R	Steuerung der Referenzmatrix (siehe Steuerung unten)
SHIFT-N	Camera an Naviculum binden

Steuerung von Camera bzw Referenzmatrix:

Linke Maustaste	freie Sicht
Rechte Maustaste	Sicht nach Vorn für interplanetare Reisen

Tasten gibt’s auch noch, sind aber nichtmehr nötig:

A, D	entlang X-Achse
T, G	entlang Y-Achse
W, S	entlang Z-Achse
Q,E	um Y-Achse
Y,X	um Z-Achse
R,F	um X-Achse

Mit der Referenzmatrix kann zum Beispiel Octree und Kollision mit dem Planetoid probiert werden.

Steuerung des Naviculums:

Maus	Ausrichtung
Leertaste	Lande-/Suchscheinwerfer ein/aus
1	wenig Schub (additiv)
2	etwas Schub (additiv)
3	viel Schub (additiv)