CG 2/3 Übungen, SS 2005

Witzmann Hannes, 932, 0126743, e0126743@student.tuwien.ac.at

Preissler Angelika, 935, 0125956, angi.preissler@gmx.at

3001 A Space Tournament

Dokumentation

Abschnitt 3

"Regelmäßig trafen sich die Nationen unserer Vorfahren, um bei den olympischen Spielen ihre Kräfte zu messen.
Nun Spielen die Planeten um die
begehrteste Trophäe der Galaxie."

(Erdenbotschafter bei der Eröffnung der ersten Marsspiele 2501)

Kurzfassung des Stroyboards

Bei „3001 A Space Tournament“ handelt es sich um ein Multiplayer Action Game, bei dem sich außerirdische Mächte treffen und einen Wettkampf veranstalten. Dabei treten immer zwei Spieler mit ihren Rover gegeneinander an.
Jeder Spieler steuert ein Fahrzeug auf dem Planeten um diverse Waffen und andere Utensilien einzusammeln. Ziel ist es mit diesen Gegenständen seinen Gegner zu schwächen und seine eigene Energie zu schützen. Nach jedem Treffer werden Punkte abgezogen. Hat ein Spieler keine Energiepunkte mehr, hat er verloren und das Spiel ist beendet.

Programmstart

Der Start des Programms erfolgt durch Aufrufen von *aSpaceTournament.exe* im Verzeichnis *\bin*.

Kurzbeschreibung des Programms

Nach Beginn des Programms gelangt man in die Startseite, von der kommt man in ein Menü, in dem zwischen verschiedenen Terrains ausgewählt werden kann (mittels den Tasten Cursor links’/’Cursor rechts’ - Bestätigung durch y’ oder mit ‚d’/’g’ - m’). Nach erfolgter Selektierung kommt die Auswahl der verschiedenen Rover, dort kann der Rover für Player 1 wie zuvor mittels den Cursor links und rechts ausgewählt werden, die Auswahl wird durch ‚m’ bestätigt. Bei Player 2 kann über die Tasten ‚d’ und ‚g’ der gewünschte Rover ausgesucht werden, bestätigt wird über ‚y’.
Diese Spieler fahren mit ihrem Rover auf einer begrenzenden Fläche und versuchen sich gegenseitig Punkte ab zu ziehen. Die Fahrt über dem Planet wird durch ein unwegsames Gelände erschwert.
Um einen Treffer zu landen müssen vorher die Waffen aufgesammelt werden, dazu muss der Rover einfach über den Gegenstand fahren. Ein Fahrzeug kann immer nur eine Waffe aufsammeln, erst nach dem Abschuss kann ein anderes Objekt aufgenommen werden. Hat der Rover bereits einen Gegenstand aufgesammelt und fährt über einen weiteren so kollidiert er mit diesem. Dieser Zusammenstoss führt zu keinem Punkteverlust, der Rover wird eventuell versetzt oder das Fahrzeug kommt ins Schleudern.

Je nach Gegenstand werden unterschiedlich viele Punkte abgezogen. Es besteht keine Möglichkeit die verlorene Energie (Punkte) wieder zu gewinnen. Am Beginn des Spiels hat jeder Spieler einen Score von 5 Energiepunkten, sobald einer 0 Punkte erreicht, hat er das Spiel verloren. Es ist auch möglich dass ein Spieler auch von seinem eigenen Gegenstand getroffen wird, was ebenfalls einen Punkteabzug zur Folge hat.

Steuerung

Tastaturbelegung:

Spiel beenden	ESC
mehr Licht	+
weniger Licht	-
Hindergrundmusik ein/aus	b
Rover- und Itemsound ein/aus	v
Wireframe ein/aus	F3
Displaylists ein/aus	F7
Transparenz ein/aus	F9

	Spieler 1 (oben)	Spieler 2 (unten)
beschleunigen	Cursor oben	r
zurück	Cursor unten	f
links	Cursor links	d
rechts	Cursor rechts	g
bremsen	m	a
Abschuss	Leertaste	y
Rover - Reset	n	w

(bei Überschlag des Rovers -> Versuch sich aufzurichten)

Spieldesign

Jeder Spieler hat am Beginn 5 Punkte. Bei jedem Treffer (auch von seiner eigenen Waffe) werden je nach Gegenstand Punkte abgezogen:

- Ball (violette Textur) → -1 Punkt, wird in die Fahrtrichtung des Rovers abgeschossen
- BallDirect (grüne Textur) → -1 Punkt, wird in die Richtung des gegnerischen Rovers abgeschossen
- Mine (grau/schwarze Textur) → -2 Punkte, wird direkt nach dem Rover abgesetzt
- Beamer (Satellitenschüssel) → Rover vom Gegner wird auf seine Startposition versetzt
- Kleine Kugel (grüne Textur) → Schutzschild, wehrt einen gegnerischen Treffer ab oder fügt bei Kollision dem Gegner einen Schadenspunkt zu.

Die verlorenen Punkte können nicht zurückgewonnen werden. Sobald ein Spieler 0 Punkte erreicht hat, hat dieser verloren und das Spiel ist beendet. Ziel des Spieles ist also dem Gegner alle Punkte zu nehmen.

Nichttriviale Objekte

- Terrain:

  Für das Terrain wurde ein Map-Editor in java implementiert. Mit ihm können Grauwert – Höhenbilder in das Spieleigene Format umgewandelt und auch bearbeitet werden. Hier können auch die Startpositionen der Spieler und die Punkte an denen Gegenstände auftauchen gesetzt werden. Leider hat die Zeit nicht mehr gereicht, um das User-Interface gut zu gestalten und die Performance zu verbessern. Im /bin Verzeichnis liegt TerrainBuilder.jar, welches den Editor startet. Wurde eine Map damit erstellt, so kann sie durch Klicken auf den Export-Button in ein Text File geschrieben werden. Eine Tabelle mit den Koordinaten jeden Vertex wird angelegt. Eine Methode der Terrain Klasse des Spieles ermöglicht das Einlesen dieser Daten aus dem Text File.

- 3D Objekte:
  Für Abschnitt 3 wurden die Modelle für die verschiedenen Rover und die Gegenstände mittels eines Milkshape Loader geladen.
  Die Modelle stammen aus verschiedenen Bibliotheken (3D Cafe, 3D Turbo Squid).

Entwurf eines Kameramodells – animierte Objekte

Das Spiel wird aus der Vogelperspektive betrachtet, es wird aber nie das gesamte Spielfeld angezeigt, sondern nur die unmittelbare Umgebung des Rovers. Die Kamera wird am unteren Rand des Spielfeldes angebracht und fährt in horizontaler Richtung mit dem Fahrzeug mit. In vertikaler Richtung wird die Verfolgung des Rovers durch einen Schwenk der Kamera nach oben bzw. unten realisiert. Da die Kamera in dieser Richtung dem Fahrzeug nicht folgt, würde dieses am Bildschirm immer kleiner werden. Um dem entgegen zu wirken, wird das Field-of-View immer dementsprechend angepasst.
Die gleichzeitige Anzeige beider Rover wird durch einen Split-Screen realisiert. Die obere Ansicht ist dem Spieler 1 zugeordnet und die untere Ansicht dem Spieler 2.

Implementierung von Texture Mapping

Das Terrain, die zwei Rover und die Gegenstände werden texturiert. Die verwendeten Texturen haben das Format BMP. Wenn ein Gegenstand abgeschossen wird, erhält er eine rote Textur.
Beim HUD und anderen teilweise transparenten Bildern wurde das TGA Format eingesetzt.

Beleuchtung/Materialien

Beleuchtung erfolgt durch die Scheinwerfer (spotlights) der Rover und einem schwachen direktionalen Lichteinfall. Die spotlights sind mittels eines Shaders implementiert um den Lichtkegel per-Pixel darstellen zu können. Es besteht die Möglichkeit die gesamte Beleuchtung des Spiels zu erhöhen bzw. zu verringern.
Das sandige Terrain besteht aus einem hoch diffusen Material. Die Items und Rover dagegen sind etwas glänzender gestaltet.

Transparenz-Effekte

Das HUD ist teilweise transparent. Ebenso die Energieanzeige, die im HUD angezeigt wird.

Außerdem ist das Energieschild halbtransparent dargestellt, um den Effekt zu erzielen, als würde eine Energiekugel den Rover umhüllen.

Ein weiterer Transparenzeffekt ist bei dem aufgewirbelten Staub zu sehen. Verschiedene teilweise transparente Texturen erzeugen den Effekt einer Rauchspur.

Experimentieren mit OpenGL

Mittels der Taste F3 kann in den Wireframe-Modus geschalten werden, um die Komplexität der Szene zu sehen. Mit F7 werden die Displaylists für Gegenstände und das Terrain deaktiviert. Die Transparenzeffekte der Rauchspur können mit F9 deaktiviert werden.

Spezialeffekte

Partikelsystem

Die Rover wirbeln bei höherer Geschwindigkeit Staub auf. Dieser Staub ist durch ein Partikelsystem realisiert. Die Textur und Transparenz ändert sich mit dem Alter des Partikels. Außerdem bewegen sich die Partikel nach oben und leicht in die Richtung, in die sich der Rover bewegte, als er das Partikel aufwirbelte.

Shader

Um die Grenzen der Lichtkegel der Scheinwerferlichter sauber darzustellen, wurde ein glsl Shader geschrieben. Dieser berechnet per Pixel den Abstand zur Lichtquelle und stellt den Kegel dementsprechend dar.

Sonstige Besonderheiten

Physik

Für die Simulation von physikalischen Eigenschaften der Rover wurde ODE verwendet. Das Fahrgestell ist mittels Joints mit den Reifen verbunden.

Auch die Collision Detection wurde durch die Open Dynamics Engine realisiert, wobei Collision Primitives wie Kugel, Box und Ebene eingesetzt wurden.

FileLoader

Um diverse Optionen, wie Auflösung, Tastaturbelegung, Rovereigenschaften und dergleichen bequem aus Textdateien einlesen zu können, wurde ein eigenes Dateiformat entworfen. Die Klasse OptionIO.cpp kümmert sich dann um das einlesen dieser Daten.
Auch für das Terrain wurde ein eigenes Format verwendet, das von dem Map-Editor erstellt und vom Spiel eingelesen wird.

Sound

Für den Sound wurde FMOD verwendet. Bei Spielstart wird während des Menüdurchlaufs ein Sound abgespielt, beim Spiel selbst gibt es einen Hintergrundsound und verschiedene Audiofiles für Motorgeräusch der Rover, Abschießen diverser Items und bei Treffer durch die Gegenstände bzw. wenn zwei Rover gegeneinander fahren.

Tools für die Modellierung der Objekte

Die Modelle stammen aus verschiedenen Bibliotheken (3D Cafe, 3D Turbo Squid), sie wurden teilweise mit Milkshape 3D, Maya und 3ds max modifiziert.

Spezielle Files

In dem Ordner data befinden sich Dateien, in denen verschiedene Optionen geändert werden können:

- display.opt
  In dieser Datei sind die Anzeigeeinstellungen (Auflösung,…) festgelegt.
- detail.opt
  Hier können Spieldetails wie Shader und Partikelsystem aktiviert und deaktiviert werden.
- controlsX.opt
  In diesen Dateien kann die Tastenbelegung der Spieler geändert werden.

Zusätzliche Libraries

- GLUT: http://www.opengl.org/resources/libraries/glut.html
- ODE: http://ode.org
- FMOD: http://www.fmod.org
- GLEW: http://glew.sourceforge.net

Quellen

http://www.lighthouse3d.com/opengl/glut/

http://www.codeworx.org/opengl_tuts.php

http://nehe.gamedev.net

http://www.glprogramming.com/red/chapter03.html

http://www.robthebloke.org/opengl_programming.html#8

http://www.3dcafe.com

http://www.turbosquid.com

http://www.lighthouse3d.com/opengl/glsl/

http://www.3dshaders.com/

http://www.opengl.org/resources/tutorials/advanced/advanced98/notes/notes.html