Spacetrash

Vasiljevs Michael 0727773 932 michael.vasiljevs@student.tuwien.ac.at
(Dempsey David 0426143 532 e0426143@student.tuwien.ac.at - bevor die zweite Abgabe)
(repository path https://svn.fsinf.at/spacetrash/trunk)

Diese Angabe enthält grosse Menge von Umformungen von Spiel Struktur/Engine. Objects können mit FBO gerendert werden (falls den Effekt eingeschaltet wird). Spiel hat 3 Stuffen. Procedurale Modellen für Asteroiden. Werwendung von Splineinterinterpolation Schiff/Sammler bewegung.

Gameplay

Steuerung des Schiffes funktioniert rein über die Mouse. Die Ziel ist einfach genug Müll zu sammeln und damit auch die Asteroide zu vermeiden, und die Mouse liefert dafür genugende Aktion mit einfache bewegung.

Effekte

Die Game hat beschränkte menge der effekten die wir seit die erste Abgabe implementieren wollte. Es gibt die folgenden Effekten:

- **Motion Blur** die Durchschitt Wert der letzten N Frames wird zusammen gespeichert.
- Erste Variante war gerade von der Red Book, die mit Accumilation gemacht ist. Accumulation Buffer und andere Pixel Operations sind realiv langsam (habe ich getestet) in vergleich zu Texturierung und Buffer Objects und das habe ich versucht zu verwenden. Die menge von tutorials haben und Forums and Gamedev.net haben mir geholfen, sodass ich FBO zu verwenden entschident haben (http://www.gamedev.net/reference/programming/features/fbo2/, die Coden aber sind schwer verandernt und in keine Fall rein copiert) Jeder Frame wirdin in FB Object gespeichert der mit Textur Object verknupft ist. Wärend Anzeigezeit, wird die Menge der gerenderten Frames mit Multitexturing an ein Quad abgedeckt, der in Parallel Projektion/Ortho2D gerendert wird. Quad wird hinter der UI gezeichnet. Pixel Shader bekommt das Array der Samplers von Texture Units, wobei jeder TU ein Frame beinhaltet, und berechten Mittelwert, Mittelvert bekommt ein Endfrabwert.
- Accumulation buffer abe wird für **Fade In/Fade out** benutzt, weil es einfache reicht die gerenderte Frame in Der Buffer zu rendern und dann ein Anteil der Farbwert zuruck zu setzen - die Anteil liegt in [0, 1] Bereich, wobei 0 ist Schwarz und 1 ist die unverändert Bild. Framebuffer Objects werden benutzt - Jedes frame wird in FBO gerendert und dann als Textur bildshirm-breites Quads (GL_QUAD) benuzt werden.
- Von Effekten die GPU Programs benutzen wurde **Per Pixel Phong Lighting** umgesezt. Denen habe ich habe mit der Hilfe der Orange Book und auch Lighthouse 3D (http://www.lighthouse3d.com/opengl/glsl/index.php?dirlightpix) tutorial geschrieben. Hierbei werden statt die Pixel Werte die Normalen interpoliert und in Fragment Program gerechnet. Für Specular Shader benuzt (View . LichtQuelle) term, statt (Normal . HalfVektor) weil es mehr warheitsgemäss und nicht viel mehr belastend an der GPU ist.
- Von anderen Effekten wurde **Statische Levels of Detail für Modellen** umgesetzt; ähnlich zu Mipmapping der Texturen, werden Objekten mit hochere Distanz von Betrachter - niedriege Modelversionen gemapt werden. In dieses fall Distanz wird vereifacht - eifach die relative Z Wert wird benutzt, weil das Object ändert in Z richtung in Beziehung zu XY Ebene mehr. Für Modellen, die proceduraleweise generiert wurden, ist es einfacher, weil nur den Parameter geändert wird, aber für importierte Modellen wäre es problematisch (auch alle Fläche-Reduzierungsfunktionalitäten von Programmen die ich gefunden habe sind dafür nicht leistungfähig genug). Ich habe entschiedet Nur Low-detail version zu wermenden - ein Tetrahedron, wobei die Unterschied nicht so detuclih erkennbar. Dafür gibt es drei Modelversionen für Nah-, Mittel- und Weitentfernung

OpenGL Betriebsarte

Andere Funktionalitäten des Spiels - OpenGL Rendermodes, es gibt: Wireframe, Immediate und Vertexbuffer Objects (könnten mit mit 'r' Taste getrieben werden). Es gibt auch FPS, Frames pro Sekunde Zeiger, der mit der Taste 'f' sichtbar geworden werden kann. FPS Zahl ist nicht so genau weil die Laufzeit jeder Frame gerechet wird (1.0 Sekunde / Ablaufzeit). Für Zeitvermessung wird High Resolution Timer Class von Song Ho Ahn benuzt (http://www.songho.ca/opengl/).

Modellen

Modelen Werden mit `CMesh` Class geladet. Es generiert sowohl procedurale Modellen als auch ladet Mesh von 3DS Dateien.
Ladung ergoflt folgendermassen:

1) Initialisirung von Parametern laut Modeltyp, Grosse und Level of Detail
2) Gestaltung von Knoten - gespeichert in STL Vector container
3) Verknufung von denen in Flächen, - gespeichert in STL List container
4) Erzeugung von Face Normals

5,6) Verknüpfung von von Knoten nach Flächen und, und erzeugung von Vertex Normals

7,8) Erzeugung von Vertex Arrays, und dann von Vertex Buffer Objects.
Das is Beschreibung von `CMesh::buildMeshData()`.

Manche Modellen für Trash abstammen von Turbo Squid (www.turbosquid.com) Portal. Procedurale modellen werdin sowohl fur Trash als auch für asteroiden werden. 3DS Loader Teil des Classes entsteht von Tutorial Loader (http://www.spacesimulator.net/tut4_3dsloader.html).
Es gibt keine selbst modelierte Objekten.
Asteroids wurden mit Rekursive Aufteilung der Ikosaeder erzeugt (Red Book, K2, Drawing Geometric Objects). Obere Flache werden mittels Kombinationen von `sin` und `cos` Funktionen mit Amplutide und Phase Abweichungen umrechnet, womit sie mehr naturlich aussehen könnten.

Interaktionen

Jedes Model bekommt ein (möglichst klein) Bounding Box in Ursprungkoordinaten und Radii der Bounding Sphere. Kollision zwischen zwei Objecten wird mit diesem Radii gerechnet. (keine Kollision wenn Distance(position2 - position1) > radii1+radii2 ist).
Für Kollisionen mit Auffänger Schiff werden auch Bounding Box Koordinaten verdendet - wenn Radii Test fehlt - wird noch Bounding Box des Objekts verwendet.
Trash Item wird es gegen den Asteroid, der mit ihm im gleichen Zeitraum in der Liste eingefügt wird für Kollision getestet.
Bewegung vom ausseren Teil des Schiffes hat den selben Algorithmus, wie von Abgabe 2. Innere Teil gemeinsam mit Camera bewegen entlang BSpline Pfad in XY Ebene, `PointsClass` Classe wird dafür verwendet. Object enthält immer Erste drei Punkten von jeweiligen Ursprung und drei Punkten von Ende des Pfads für Kontinuität. Class ist total Selbst implementiert mit der Hilfe des 6.3 Kapitels - B-Spline Curves, 3D Computer Graphics (Alan Watt).

Bibliotheken

SDL für fenstersteuerung
SDL_image - Bilder von bmp, png, jpg Formaten.
SDL_mixer - Soundeffekten
FTGL und davon abhängige FreeType Bilbiothek für text Rendern
GLee wird für Windows Kompatibilität werwendet.

Verbesserungen/Effekten die fehlen

Ich würde gerne die volgende Effekten machen:

- Bump Mapping und/oder Alternativen wie Normal, Parallax Mapping,
- Statt Bitmap Font fon GLUT - FTGL Texture Font
- Animierte teil: Schiff Rohr, mit VBO/Vertex arrays gezeichnet - `GL_DYNAMIC_DRAW`, in vergiech zu `GL_STATIC_DRAW` für statische Objekten.

Zu viel Zeit wurde and Shiff Bewegung verwendent, das ist einbisschen übertrieben - zwei Zustanden würde reichen (statt Splines).

Spielen

Spiel hat noch nur drei Stuffen (aber sie können erweitert werden). Erste stuffe ist eifach - sammle was man sieht. Zweite Stuffe Sind mit Asteroiden - mehr Spassig (hoffe ich :)). Dritte stuffe ist Freitod Mission - der Hinweis hier ist um die Rote Powerups zu sammeln sodass man die Asteroids auch sammeln kann.