

PANG 3D

(3. Abgabe)

Peter Bubestinger
Florian Kraner

0025993
0025583

Für Abgabe #3 wurden folgende Dinge implementiert/erweitert:

- Spieler
- Keyboardsteuerung
- Beleuchtung
- Materialien
- Gegner (Blasen)
- 2D Text
- Komplexe Models (.3DS Files)
- Gameplay (Lives, Score, GameOver, ...)

Keyboardsteuerung:**Tastenbelegung (Spiel)**

| | |
|--------------------|--|
| 'D' | Toggle Lock Mode |
| ,F' | Jump to next lock |
| 'Space' | Shoot |
| 'Left/Right Arrow' | Strafe Left/Right – or rotate Left/Right |
| 'Up/Down Arrow' | Move Forward/Backwards |
| 'Esc' | Exit Game |

Tastenbelegung (Debug)

| | |
|---------|--|
| ,+' | Erhöht den Score |
| ,-' | Zieht Leben ab |
| ,L / I' | Beleuchtung an/aus |
| ,g' | Toggle Kamera Goto-Modus |
| 'x' | Toggle Kamera Target-Modus |
| 'r' | Toggle Kamera RotateAroundTarget-Modus |
| 'c' | Toggle Kamera RotateAroundCenter Modus |
| 'n' | Toggle Normalize Hero.RotationSpeed |

Tastenbelegung (Debug 2)

| | |
|------|-------------------------------------|
| 'F1' | Hilfe |
| 'F2' | Toggle Anzeige von FPS |
| ,F3' | Toggle WireFrame Modus |
| 'F8' | Toggle FrustrumCulling (invertiert) |

Beim FrustrumCulling-Modus werden jedoch die normalerweise „nicht gezeichneten“ Polygone dargestellt (um den Effekt zu verdeutlichen).

Spieler

Der Spieler wurde sowohl in Hinsicht auf Graphik als auch Steuerbarkeit wesentlich verbessert. Statt dem zu Testzwecken verwendeten Würfel, ist es nun ein kleines Raumschiff mit Textur und Antriebsanimation.

Wir haben auch versucht, die bereits bis zur 2. Abgabe entwickelten Features der Kamera besser auszunutzen, was sich im ‚bewegteren‘ und flüssigeren Spieleablauf widerspiegelt.

Eine große Herausforderung für uns war es, die Steuerung des Spiels im Allgemeinen zu verbessern, und wir entschieden uns für 2 unterschiedliche Bewegungs-Modi:

1) Arena-Lock Modus

Jede Arena (in unserem Fall gibt es zwar nur eine, aber zukünftig könnten es mehr werden) hat eine gewisse Anzahl an Achsen, zu denen man sich annähernd parallel bewegt. Bei unserer quadratischen Arena gibt es 8 Achsen: 4 durch die Eckpunkte und 4 parallel zu den Kanten.

In diesem Modus kann man sich schnell drehen und somit hakenartige Ausweichmanöver durchführen.

2) Bubble-Lock Modus

Wenn der Bubble Lock Modus aktiviert ist, richtet sich die Bewegung des Spielers an der jeweilig ausgewählten Blase aus (welche mit einem Fadenkreuz-artigen, grünen Ring markiert wird). Das heißt, dass man beim nach-vor-gehen sich direkt auf die selektierte Blase zu bewegt, und sich mit links/rechts um die Blase dreht.

Dieser Modus eignet sich besonders dazu, eine bestimmte Blase zu zerschießen, da man sich mit Hilfe der Rotation sich der Blase leicht in den Weg stellen kann.

Externe Programmbibliotheken: (selbe wie bei 2. Abgabe)**1) Glut32**

Zur Darstellung eines Windows-Fensters, sowie zur Auswertung gewisser Ereignistasten (Bewegungssteuerung wurde aus Performancegründen direkt über Windows-Funktionen implementiert)

2) DevIL

Zum Laden von Texturen aus externen Bilddateien.
Folgende Funktionen wurden verwendet:

```
ilGenImages  
ilBindImage  
ilLoadImage  
ilGetError  
ilGetInteger
```

Diese dienen lediglich dem Laden eines Bildes. Sämtliche Schritte, diese Bildinformation als OpenGL-Texture zu verwenden werden mit OpenGL Befehlen durchgeführt. (d.h: NICHT mit „ilutGL...“ Funktionen von DevIL)

3) 3D-Studio Model-Loader

Beziehbar unter http://www.gametutorials.com/Tutorials/opengl/OpenGL_Pg4.htm stammen die Klassen zum Einlesen des Formats aus diesem Tutorial. Es wird darin des Weiteren eine Datenstruktur gegeben, die wir verwendet haben um die OpenGL-Befehle zur Darstellung hierauf wiederum in unserem Code zu implementieren.

Spezialeffekte:

1) Transparenz

Die meiste Transparenz wird mit Hilfe von unterschiedlichen Blending-Funktionen erreicht, die schwarze und dunkle Bildinformation als jeweils unterschiedlich durchsichtig erscheinen lassen, wodurch ein höherer Realismus erzielt wird als bei herkömmlichen Alpha-Funktionen. Alpha-Information verwenden wir z.B. beim Markieren von Bubbles, oder beim Anzeigen von Schrift-Sprites um diese nach Bedarf auszuschneiden. Transparent sind die Bubbles, die Schatten, der Schuß und der Rauch unter dem Schiff.

2) Display-List

Objekte werden beim Zeichnen generell aus der Display-list geladen. Von Ein/Aus-Schalten haben wir erst später gelesen und es erschien uns sinnlos zusätzlichen (verhältnismäßig großen) Aufwand zu betreiben nur um die FPS wieder zu SENKEN, es gab also andere Dinge die uns wichtiger waren.

3) Display-List

Objekte werden beim Zeichnen generell aus der Display-list geladen. Von Ein/Aus-Schalten haben wir erst später gelesen und es erschien uns sinnlos zusätzlichen (verhältnismäßig großen) Aufwand zu betreiben nur um die FPS wieder zu SENKEN, es gab also andere Dinge die uns wichtiger waren.

4) Partikel-Effekt

Beim Abschuß einer Bubble wird diese nicht nur gespalten, sondern es bilden sich auch mehrere kleine zufällige Ableger die dem Boden entgegenschweben.

5) Bewegter Rauch

Wahrscheinlich anders als erwartet, befindet sich unter dem Schiff eine Rauchspur, die aus mehreren Texturen besteht um einen natürlicheren Eindruck zu vermitteln.

6) Schatten

Aus Gründen des Gameplays unverzichtbar, befinden sich exakt unter den Bubbles gerade projizierte Schatten. Da diese eine wichtige Orientierungshilfe sind dürfen sie sich nicht bewegen und somit muss leider auch unsere Lichtquelle starr bleiben.