

Bulb Fiction

CG2 LU SS08

Spielinformation und Steuerung

Barney Bulb muss sein Kraftwerk wieder aktivieren, dazu muss er zwei Rätsel lösen.

Nach dem Programmstart kann man im Menü Auflösung und Farbtiefe durch Klicken auf die jeweiligen Monitore ändern, sowie Lautstärke für Effekt und Musik einstellen. Ein Klick auf „Start!“ lädt das Spiel.

Rätsel 1:

Ziel des Spiels ist es, das Licht der Spielfigur über fünf optische Bänke auf die Solarzellen an den Enden der Ausleger zu leiten. Dabei sollen die Zellen möglichst voll ausgeleuchtet werden. Außerdem muss das Licht aus der Glühbirne vollständig genutzt werden. Weitet eine Linse den Lichtkegel zu sehr, sodass dieser nichtmehr vollständig durch die nächste Linse passt, geht Intensität verloren.

Die an der Solarzelle gemessene Gesamt-Lichtintensität wird am unteren Bildschirmrand in Prozent angezeigt. An allen fünf optischen Bänken müssen 100% erreicht werden, um das Puzzle zu meistern, dabei dreht sich der Kreisel aber immer schneller, je mehr Bänke bereits richtig ausgerichtet sind.

Steuerung für Rätsel 1:

L	Licht einschalten
Q, W, E, R, T	Bank auswählen
1, 2, 3, 4, 5	Linse auswählen
Pfeiltaste rauf/runter	Linse verschieben
,/.	Zoom +/-
Maus X-Richtung	Kamera um ausgewählte Linse rotieren
Maus Y-Richtung	Kamera Höhe
X	Reset auf Ausgangsstellungen der Linsen
C	Cheat: alles auf die richtigen Positionen setzen
Esc	Zurück ins Menü

Rätsel 2:

In diesem von „Zelda“ inspiriertem Rätsel müssen die zwei auf den Feldern sitzenden Objekte auf die beiden roten Felder gebracht werden. Dabei springen sie immer in die Richtung, in die sich die Spielfigur dreht und springt. Gesteuert wird dabei nur mit den Cursortasten. Blickt die Spielfigur geradeaus und drückt man die rechte Cursortaste, dreht sich die Spielfigur nach rechts. Dasselbe tun die beiden Objekte, von ihrer Sicht ausgesehen. Die Blickrichtung der Spielteilnehmer wird durch kleine Kugeln angezeigt. Um jetzt auf das nächste Feld zu springen muss dieselbe Taste wieder

gedrückt werden. Drückt man eine andere Cursortaste dreht sich die Spielfigur wieder in die entsprechende Richtung. Zur Vereinfachung der Spielsteuerung bleibt die Kamera immer an derselben Position. Das Rätsel wird erfolgreich beendet, wenn man die zwei Objekte auf die 2 roten Felder gebracht hat, wobei es egal ist welches Objekt auf welchem roten Feld landet.

Steuerung für Rätsel 2:

Pfeiltasten	Drehen und springen
X	Reset auf Ausgangsstellungen
Esc	Zurück ins Menü

Anforderungen:

Gameplay

Gameplay ist natürlich vorhanden, zwei Rätsel müssen gelöst werden. Man bekommt Feedback über die getätigten Aktionen über ein HUD (Prozentanzeige), sowie über Audioausgabe (Fanfaren-Sound bei erfolgreichem Abschluss).

Nichttriviale Objekte

Neben einfachen texturierten Flächen, wie Wände und Böden werden MD2-Models für komplexere Strukturen eingesetzt. Diese sind neben der Glühbirne die Batterien im 2. Rätsel, das Gestänge für die Linsen, die Linsen selbst und das Gelände um den Lichtkeisel. Der Model-Loader basiert auf dem Code von Allen Sherrod (www.ultimategameprogramming.com) und wurde um eine Berechnung der Vertex Normals erweitert. Die Models wurden in Cinema4D erstellt, die PCX-Texturen mit GIMP erzeugt. Anschließend wurde das Model in Misfit Model 3D mit den PCXs texturiert, animiert und als MD2 abgespeichert.

Andere nicht-triviale Objekte sind die Lichtkegel. Dazu wurde eine Funktion implementiert, die eine Hauptachse, zwei Mittelpunkte und zwei Radien übergeben bekommt und daraus einen transparenten Lichtkegel mit Helligkeitsverlauf generiert. Ein nützlicher Nebeneffekt ist die Entstehung eines Brennpunktes wenn der zweite Radius negativ ist.

Animierte Objekte

Das Glühbirnen-Model ist animiert (springt auf und ab) und nutzt eine framebasierte Animation mit zwei Keyframes (Springen-unten, Springen-oben), die Zwischenframes werden je nach Framerate berechnet.

Beschleunigung der Sichtbarkeitsberechnung

View Frustum Culling fehlt leider noch komplett und wird hoffentlich bis zum Spieleevent noch implementiert. Eine kleine Optimierung ist aber die Nutzung von Backface-Culling.

Transparenz-Effekte

Transparenz wurde für die Lichtkegel verwendet. Der Helligkeitsverlauf beginnt bei einem Alpha-Wert von ca. 50% und endet bei 0%.

Experimentieren mit OpenGL

F2	Framerate anzeigen
F3	Wireframe an/aus
F7	Display Lists an/aus (nur im 1. Rätsel)
F9	Transparenz an/aus (nur im 1. Rätsel)
F10	Fragment Shader an/aus (nur im 2. Rätsel)

Spezialeffekte

In Rätsel 1 wurde eine Reflexion mittels Stencil Buffer implementiert. Der Basiscode stammt aus dem Reflection Tutorial auf <http://nehe.gamedev.net>. Der Code beinhaltete ursprünglich nur eine Spiegelung einer Kugel über einer rechteckigen Ebene. Der Boden wurde durch eine kreisförmige Ebene ersetzt, in der sich die gesamte Szene des 1. Rätsels spiegelt.

In Rätsel 2 wurde Per-Pixel Lighting im Fragment Shader implementiert. Der Code dazu basiert ebenfalls auf dem entsprechenden Tutorial auf <http://nehe.gamedev.net>, auf dem Wissen aus der CG1 VO/LU, sowie dem Tutorial aus dem eingesetzten CG Toolkit. Die Lichtquelle befindet sich in der Mitte der Szene über dem Spielfeld.

Die Lichtquelle im 1. Rätsel befindet sich übrigens (logischerweise) im Zentrum der Glühbirne, deswegen sieht die runde Oberfläche der Birne aus wie flat-shaded. Ansonsten wäre die Birne schwarz, da das Licht von innen kommt.

Sonstige Features

- Es gibt ein Menü für Graphik- und Soundeinstellungen. Somit erspart man sich das Erstellen eines Textfiles mit GameMode-String.
- Die meiste Zeit für den Programmieraufwand für Rätsel 1 brauchten wir für die Berechnung der Lichtbrechung. Sie entspricht zwar noch immer nicht 100% den realen Bedingungen, ist aber schon fähig Sammel- und Streuungs-Linsen zu berechnen. Dazu wurde ein Structure „Lens“ erstellt, die einen Wert „Refraction“ enthält. 1.0 bedeutet, dass die Linse das Licht ungebrochen durchlässt. Ein Wert < 1.0 steht für eine Sammellinse, > 1.0 ist eine Streuungslinse. Zu jeder Linse wird auch die empfangene Menge an Licht gespeichert. Passt

der Lichtkegel in die Linse ist der Wert 1.0, ist er breiter verringert sich der Wert, je nachdem wie viel Licht auf der Seite verloren geht.

- Im 2. Rätsel sind die Sprünge teilweise animiert, funktionieren aber derzeit leider nur beim Springen in Richtung negativer z-Achse.
- Sämtliche Bewegungen sind framerateunabhängig implementiert. Die Framerate wird alle 5 Frames neu berechnet und an die OpenGL-relevanten Klassen weitergegeben.
- Die Glühbirne bekommt eine hellere Textur wenn sie leuchtet.
- Rätsel 1 beinhaltet ein HUD, auf dem alle relevanten Helligkeitsinformationen gezeigt werden.
- Bis auf die animierte Glühbirne werden sämtliche Models und Teile der Umgebung mit Display-Lists gerendert. Diese Funktion kann mit F7 ausgeschalten werden. Auf einem aktuellen PC mit Geforce 8 bricht die Framerate dann im 1. Rätsel von 200 auf ca. 20 FPS ein.
- Hintergrundmusik

Libraries

- GLUT
- FMOD
- CG Toolkit für Per-Pixel Lighting
- DevIL für Texture Loading