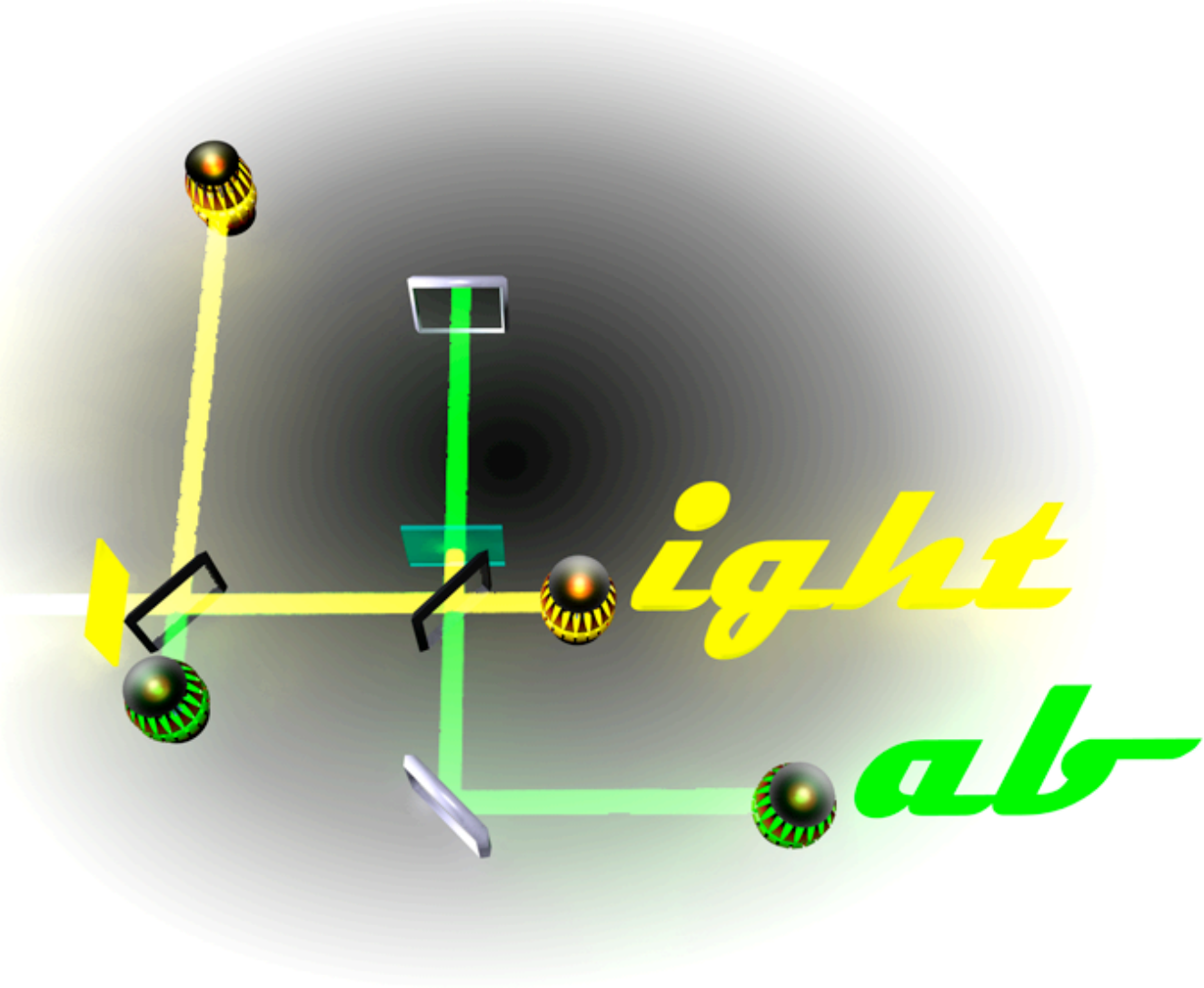
lightlab

Dokumentation (1. Endabgabe (Nachabgabe), 26.06.2009)



CG2LU, SS2009

Bauer Stephan, 033532, 0626835, stephan.bauer@student.tuwien.ac.at
Gradisnik Thomas, 033532, 0626980, e0626980@student.tuwien.ac.at
Zalesak Wolfgang, 033532, 0727115, zalesak.wolfgang@gmx.at

Spieltyp

Puzzle

Ziel des Spiels

Von Laseremittern ausgehende Laserstrahlen müssen in bestimmte Ziele (und Zwischenziele) gelenkt werden, dabei ist darauf zu achten, dass die Strahlen richtige Farben aufweisen, wenn sie ein entsprechend definiertes Ziel erreichen. Es stehen mehrere Elemente zur Verfügung, die einen Laserstrahl beeinflussen können (Spiegel, Farbfilter, Kollektoren, ...) und müssen entsprechend platziert und konfiguriert werden. Die Platzierung der Elemente wird von Hindernissen beeinflusst, die sich im Labor befinden (Wände, Gegenstände, Erhöhungen, ...).

Gameplay

Alle Ziele sind durch Positionieren von Spiegeln, Filtern und Kollektoren zu treffen. Der Laser hat keine unbegrenzte Energie, den jeweiligen Energiestand beschreibt die Anzeige am unteren Rand des Fensters. Um Energie zu sparen, während Elemente positioniert werden, kann der Laser ausgeschaltet werden (Taste Power, oder Leertaste). Hat der Laser keine Energie mehr und ist der Level nicht gelöst, ist das Spiel verloren.

Kurzbeschreibung der Implementierung

Frei bewegbare Kamera und Steuerung

Durch Drücken rechten Maustaste und Bewegen der Maus kann der Blickwinkel in alle Richtungen verändert werden. Mit dem Mausrad kann die Kamera in Blickrichtung vor und zurück bewegt werden. Mit WASD kann der Blickpunkt links/rechts und rauf/runter bewegt werden, mit QE kann die Höhe des Blickpunkts verändert werden.

Platzierung und Konfiguration von Objekten

Verfügbare Objekte im Spiel sind Emitter, Kollektor, Ziel, Laserstrahl, Färbungsfilter, Spiegel und halbdurchlässige Spiegel. Dabei sind Färbungsfilter, Spiegel und halbdurchlässige Spiegel bewegbar.
Die Ausrichtung (Drehung der Objekte um zwei Achsen) der bewegbaren Objekte erfolgt per Maus durch Auswählen der Objekte mit der linken Maustaste und anschließender Bewegung (bei weiterhin gedrückter Maustaste). Die Positionierung (das Verschieben) der Objekte erfolgt durch einen Doppelklick. Dann haftet das ausgewähle Objekt am Mauseiger. Mit einem Klick kann dann das Objekt auf einer Wand positioniert werden. Sollte das Objekt zu nahe an einem anderen sein, so wird dies grafisch dargestellt (Wireframe) und kann nicht positioniert werden. Zu Debugzwecken kann auch noch das Objekt mit den Pfeiltasten und zusätzlich mit den Tasten für Punkt und Strich verschoben werden.

Features

Es stehen mehrere zusätzliche Rendering-/Debug-Ansichten zur Verfügung: Die Funktionstasten sind hierbei wie gefordert belegt:

F2 - Framerateanzeige ein/aus
F3 - Wire Frame ein/aus
F4 - Texturqualität verändern: Nearest Neighbor/Bilinear
F5 - MipMapping-Qualität ändern: Aus/Nearest Neighbor/Linear
F8 - Viewfrustumculling ein/aus

F1, F6, F7 und F9 sind nicht verfügbar.

Als zusätzliche Ansicht gibt es mit die **Boundingsphere-Ansicht** die durch Drücken von F10 aktiviert wird. Mittels F1 können die Spiegelungen deaktiviert werden.

Weitere Funktionen sind über die folgenden Tasten auswählbar:

Leertaste – Laserstrahl ein/aus
Enter – Licht ein/aus

Commandozeilenbefehle

-fullscreen
-resolution [width] [height]
-level [number]

Beleuchtung und Texturierung

Wir verwenden 3D prozedurale Texturen und 2D geladene Texturen verwendet. Der Emitter, das Ziel, der Kollektor und die Halterungen werden mit einer 3D-Texture texturiert, Wände und Rahmen mit 2D Texturen. Eine einzelne Lichtquelle mit Richtung (-1,-1,-1) beleuchtet den Raum.

Verwendete Libraries

Als Basis für den Modelloader dient eine Class von www.spacesimulator.net, der umfangreich adaptiert wurde (Berechnung von Normalen, Texturkoordinaten, ...).
Zum Laden von Bildern für Texturen und UI verwenden wir das empfohlene DevIL (http://openil.sourceforge.net/), wobei nur IL und ILU verwenden werden.
Für die Darstellung des UI verwenden wir ein adaptiertes GUIchan (eigener Imageloader mittels DevIL) (http://guichan.sourceforge.net/wiki/index.php/Main_Page). Für die Musik benutzen wir die Libraries von FMOD (http://www.fmod.org).

Animationen

Die Ziele sind animiert und Verformen sich, wenn Sie vom richtigen Laserstrahl getroffen werden (Farbe) und kehren in die Ursprungform zurück, falls die vorherige Bedingung nicht mehr zutrifft.

Spezialeffekte

Komplexe Transparenz
Laserstrahlen und Filter sind Transparent. Da Laserstrahlen additiv wirken, Filter jedoch subtraktiv – daher ist eine Sortierung erforderlich. Auf Grund der langen Laserstrahlen ist eine einfache Sortierung nach Z-Werten nicht ausreichend. Die Laserstrahlen werden entlang der Filterebenen zerteilt und anschließend wird der gesamte Raum anhand der Ebenen der Filter in Räume zerteilt. Diese werden entsprechend vor/nach einem Filter klassifiziert. Ein selbst entwickelter Algorithmus geht diese Räume intelligent durch und findet so eine optimale Sortierung der transparenten Objekte statt. Der Großteil der Berechnungen erfolgt Blickpunkt unabhängig und kann somit für Reflektionen mit geringem Rechenaufwand verwendet werden.

Planare Spiegelungen
An Spiegeln finden planare Spiegelungen statt. Dafür wird der Stencilbuffer genutzt. Die Szenen wird dann an der Spiegelebene gespiegelt neu gezeichnet. Eine rekursive Implementierung ist vorhanden und bei deaktiviertem Laserstrahl gibt es Mehrfachspiegelungen mit einer Tiefe von 2. Da die Spiegel freistehend sind, wurden mittels Clippingplanes eine korrekte Spiegelung sichergestellt.

Quellen:

http://www.scribd.com/doc/4838089/Stencil-Buffer-in-OpenGL

Per-Pixel-Phong Illumination
Im Fragmentshader wird ein Per-Pixel Lightning berechnet.

Quellen:
http://www.gamedev.net/reference/programming/features/glsllib/page5.asp
http://www.nightspawn.de/files/gltut/html/node47.html

Modelle
Die Modelle wurden mit Cinema4D modelliert.
Zum Berechnen der Texturkoordinaten wurde Blender verwendet.

Shadowmap
Aus der Sicht der Lichtquelle wird eine Shadowmap berechnet. Diese wird dann im Vertex- und Fragmentshader zur Berechnung der Schatten benutzt.

Quellen:
http://fabiensanglard.net/shadowmapping/index.php