

Incarcerated

Gameplay

Es werden an vordefinierten Punkten (jeweils ein Punkt pro Ebene) in zufälliger Reihenfolge Gegner erstellt, die anschließend versuchen den Hauptcharakter zu erreichen. Der Spieler ist mit einem vollautomatischem Gewehr ausgerüstet und muss versuchen, so lange wie möglich zu überleben. Es gibt zwei Gegnertypen, die sich einerseits optisch, als auch in der Bewegungsgeschwindigkeit und der Anzahl an Lebenspunkten unterscheiden. Für jeden getöteten Zombie bekommt der Spieler Punkte. Erreicht ein Gegner den Hauptcharakter ist das Spiel vorbei und der Highscore wird angezeigt.

Lighting, shadows and materials

Die Charaktere, die Lampen und die Böden sind texturiert, haben das gleiche Material und haben Normals. Es gibt 1 diffuse Lichtquelle („Mond“) und 3 Spotlights (Straßenlaternen), wobei unterschiedliche Lichtfarben verwendet werden. Alle Objekte in der Szene werden von der Diffusen Lichtquelle angestrahlt. Alle Objekte innerhalb des Spotlight-Cones werden zusätzlich von der jeweiligen Spotlight angestrahlt. Für jede einzelne Lichtquelle wird eine Shadow Map erstellt und im Render Pass verarbeitet, wobei der Schatten der diffuse Lichtquelle etwas heller ist als die Restlichen.

Effects

- GPU Vertex Skinning (Michael Oppitz)
Referenz: http://content.gpwiki.org/index.php/OpenGL:Tutorials:Basic_Bones_System, accessed on 22/06/2015.
Referenz: <http://ogldev.atspace.co.uk/index.html>, Tutorial 38, accessed on 22/06/2015.
- Spotlights (Matthias Reisacher)
Referenz: <http://ogldev.atspace.co.uk/index.html>, Tutorials 19-21, accessed on 22/06/2015.
- Shadow Maps + PCF (Matthias Reisacher):
Referenz: *Presentation slides der LVA*.
- Nebel (Matthias Reisacher)
Referenz: *Lengyel, Eric. Unified Distance Formulas for Halfspace Fog. Journal of Graphics Tools. vol.12, no.2, 2007.*

Complex Objects

Alle verwendeten 3D-Modelle wurden von uns selbst modelliert, texturiert und animiert. Dafür wurden die Programme Maya 2016, Zbrush 4R7 und Photoshop CC verwendet. Das importieren der Modelle wurde mit der Library Assimp 3.0 bewerkstelligt.

Animated Objects

Der Hauptcharakter, sowie die Gegner werden animiert. Es handelt sich hierbei um eine Skeletal Animation bei allen animierten Modellen. Für den Hauptcharakter gibt es Animationen für den Idle-Zustand sowie Vorwärts-, Rückwärts-, Links- und Rechtsbewegung. Das Vertex Skinning wird von der GPU angewendet.

View-Frustum-Culling

View-Frustum-Culling ist standardmäßig eingeschaltet und kann mittels Drücken der F8-Taste ein- und ausgeschaltet werden. Die Anzahl der gerenderten Modelle wird in den Renderinformationen (F2-Taste), auf der linken Seite in der 2. Zeile ausgegeben.

Transparency

Wird ein Gegner vom Spieler getötet, wird eine sterbe Animation abgespielt und der Gegner wird über einen Zeitraum von 1.5 Sekunden linear abfallend transparenter angezeigt, bis er vollständig verschwindet. Mit der F9-Taste kann dieser Effekt ein- und ausgeschaltet werden.

Experimenting with OpenGL

- Für alle Modelle werden VAOs & VBOs verwendet.
- Im Zuge von Shadow Mapping werden FBOs verwendet.
- Beim Laden der Textur erstellt OpenGL automatisch eine MipMap.
- Die Textur-Sampling-Quality kann verstellt werden (am besten an den weißen Bodenmarkierungen der Parkplätze erkennbar).

Control

Beschreibung	Tasten
Bewegung des Hauptcharakters	W, A, S, D
Rotation des Hauptcharakters	Seitliche Mausbewegung
Zoom in/out	Mausrad
Schießen	Linke Maustaste
Jetpack benutzen	Leertaste
Show Render-Infos on/off	F2
Wire Frame on/off	F3
Textur-Sampling-Quality: Nearest Neighbor/Bilinear	F4
Mip Mapping-Quality: Off/Nearest Neighbor/Linear	F5
Nebel on/off	F6
Show Bounding Boxes on/off	F7
Viewfrustum-Culling on/off	F8
Transparency on/off	F9

Features

Zusätzlich zu den Requirements wurden folgende Features bisher umgesetzt:

- Skybox (auf Grund der schlechten Texturwahl ist die Skybox nur gut erkennbar, wenn aus der Szene heraus gezoomt wird).
- Eine am Boden liegende Nebelschicht.
- Selbst geschriebener Scene Graph (rekursiver Octree).
- Selbst geschriebene Collision Detection (Cylinder to AABB).
- Streuung der Waffe. Wird nur ein einzelner Schuss abgegeben, verzieht die Kugel nur gering. Bleibt der Spieler jedoch auf der linken Maustaste (Schießen), wächst der Streuungsfaktor mit jedem Schuss kontinuierlich an, bis das Maximum erreicht ist. Das Gewehr hat 30 Kugeln im Magazin. Sind diese aufgebraucht wird automatisch nachgeladen. Während dem Nachladen kann nicht geschossen werden.
- Jetpack. Wird mit der Leertaste gestartet und bewegt den Hauptcharakter nach oben. Nach jedem Benutzen wird ein Cooldown (5 Sekunden) gestartet, während dem das Jetpack nicht verwendet werden kann.
- Selbstgeschriebene AI für die Gegner.

Additional Libraries

Folgende externe Libraries wurden im Zuge dieser LVA bisher von uns implementiert:

- Assimp 3.0 - <http://assimp.sourceforge.net/>
- FreeImage 3.16.0 - <http://freeimage.sourceforge.net/>
- GLEW 1.12.0 - <http://glew.sourceforge.net/>
- GLFW 3.1.1 - <http://www.glfw.org/>
- GLM 0.9.6.3 - <http://glm.g-truc.net/0.9.6/index.html>