

Escape from Exocron

Adis Malkic
Christoph Patscheider
Christoph Payrhuber

Idee
Peter „ph0x“ Moraliyski



1. Spielbeschreibung

1.1 Storyline

Der Spieler befindet sich zu Beginn des Spiels in einer kleinen Stadt namens Zlin auf dem Planeten Exocron. Der Kern des Planeten wurde durch ein fehlgeschlagenes Kernfusionsexperiment destabilisiert. Die dafür verantwortliche Forschungsfirma Umbrella Corp. Inc. hat daraufhin den gesamten Planeten evakuiert. Der Spieler wurde durch das fehlgeschlagen Experiment auf der Zeitlinie versetzt und findet sich nun nach der Evakuierung des Planeten wieder.

Durch seine Beschäftigung bei der Umbrella Corp. Inc. ist ihm bekannt, dass in der nächstgelegenen großen Stadt Niew ein defekter Raumgleiter stehen müsste, dessen Reparatur er bewerkstelligen könnte. Einziges Problem auf seiner Reise nach Niew sind die alten und noch immer scharf geschalteten Verteidigungsanlagen des Planeten. Ausgestattet mit einem einfachen Schild zum Schutz muss er die Reise wagen...

1.2 Start und Steuerung

Befindet man sich im Verzeichnis des Spiels kann dieses mit folgendem Befehl gestartet werden: `../escape -f stereo-cave.csm`.

Der Spieler wird durch das Spiel geführt und muss die auf ihn zukommenden Schüsse mit einem Schild abwehren. Durch die Ausrichtung der Wand und der Brille mit Tracking Sensor wird das Schild platziert. Die Wand ist als Griff für das Schild zu sehen. Somit werden alle Bewegungen der Wand direkt auf das Schild übertragen.

Wand-Knöpfe:

- Das Spiel wird mit Button 1 der Wand gestartet.
- Mit diesem kann der Spieler das Spiel auch pausieren.

Konsole:

- „ESC“: Spiel beenden
- „Space“: Spiel starten/pausieren
- „i“: Unverwundbarkeit ein/ausschalten

1.3 Highscore-System

Auf der frontalen Projektionswand wird dem Spieler in der linken oberen Ecke sein momentaner Punktestand angezeigt. Der Punktestand wird bei Erreichen von bestimmten Wegpunkten um 10 Punkte erhöht. Wehrt der Spieler einen Schuss mit dem Schild ab, wird der Punktestand ebenfalls um 10 Punkte erhöht. Wird der Spieler von einem Schuss getroffen, werden hingegen 10 Punkte abgezogen. Ein Treffer hat auch den Abzug von 10 Lebenspunkten zur Folge. Der Spieler hat am Start 200 Lebenspunkte zur Verfügung. Die Lebenspunkte werden rechts oben auf der frontalen Projektionswand angezeigt. Sinken die Lebenspunkte auf Null, ist das Spiel beendet.

2. Technischer Hintergrund

Das Spiel wurde in C++ entwickelt. Zur Entwicklung wurden verschiedene frei verfügbare Bibliotheken verwendet. Im Folgenden ein kurzer Überblick über die eingesetzten Bibliotheken, Tools und Technologien.

- Szenengraph: Als Szenengraph wird OpenSG verwendet. Zusätzlich wird noch der auf OpenSG basierende CAVE Scene Manager (CSM) eingesetzt, der die Konfiguration des CAVEs und andere CAVE-spezifische Funktionalitäten bereitstellt.
- Skybox: Zur Darstellung des Hintergrundes eines Levels wird eine Skybox eingesetzt. Dabei wird auf die von OpenSG zur Verfügung gestellten Skybox-Funktionalitäten zurückgegriffen.
- Audio: Als Basis für die Wiedergabe von Musik und Soundeffekten wird OpenAL verwendet. Die Soundeffekte werden in Abhängigkeit von der Position der Sound-Quellen und des Spielers in unterschiedlicher Lautstärke wiedergegeben. Als Audio-Format für die Hintergrundmusik wird OggVorbis unterstützt. Zum Decodieren der OggVorbis Dateien werden die von Xiph.org zur Verfügung gestellten Bibliotheken libogg und libvorbis benützt. Die Decodierten Audiodaten werden per Streaming über OpenAL-Queues abgespielt.
- 3D-Modelle: Die 3D-Modelle wurden in Maya 2008 erstellt. Das Schild, die Türme und Gebirge des Levels wurden vollständig in Maya erstellt. Weitere Modelle welche sich in der Welt finden, stammen von diversen Internetseiten (hauptsächlich <http://www.max-realms.com>). Aufgrund fehlender Importfähigkeiten von Maya wurde Blender verwendet, um die .3ds Modelle in ein Maya-kompatibles Format zu konvertieren. Aus Maya wurden die Modelle ins VRML-Format exportiert. Dieses

wurde bei komplexen Modellen aus Performancegründen noch in das OpenGL-native Format .bin konvertiert.

- Level-Daten: Die Informationen zum Aufbau eines Levels werden in einer XML-Datei gespeichert. Im Konkreten sind das Pfad und Transformationsinformationen der zu ladenden Modelle. Dadurch ist die Erstellung eines neuen Levels stark erleichtert. Zum Laden und Parsen der XML-Datei wird irrXML verwendet.

2.1 Problembereiche

Die anfänglich größte Hürde bei der Entwicklung des Spiels war, dass zwei der drei Gruppenmitglieder keine Erfahrung mit C++ hatten. Dies bedurfte Einiges an Geduld und Einschulungsaufwand des restlichen Teams (Danke an dieser Stelle).

Probleme stellten ebenfalls die Kollisionsabfrage zwischen Schild und Schuss, die Einstellung der Soundeffekte (abhängig von der Position des Spielers) und die Vielfalt der 3D-Formate dar. Diese konnten teilweise durch das Finden der Ursachen bzw. durch Workarounds gelöst werden.

3. Screenshots







4. Quellenverzeichnis

Modelle: <http://www.max-realms.com>.

Soundeffekte: <http://www.fan-files.com>.

Skybox-Textur: <http://www.fpsbanana.com>.

Hintergrundmusik: Zentriert ins Antlitz: E.M.I.T.R – Colours, Track 1: Silicon Worlds,
<http://www.zentriertinsantlitz.de>.