

BatteryRun – Submission 2

Das Spiel ist fertig implementiert. Man kann sich durch die Welt bewegen, das Kabel einsammeln und dann nachhause gehen.

Gameplay

3D Geometry

Es werden Objekte aus externen Dateien geladen, die dann in der Welt platziert werden.

Als Importer wird Assimp¹ verwendet.

Win/Loose Condition

Das Spiel ist gewonnen, wenn das Ladekabel eingesammelt wurde und man sich dann zur Haustür bewegt.

Verloren hat man, wenn man sich aus der Welt hinausbewegt oder ins Wasser fällt.

Controls

Die Steuerung erfolgt mit den Tasten WASD um zu gehen und die Maus um die Blickrichtung zu ändern.

Die Tasten ändern dabei die Position der Kamera und die Maus die horizontalen und vertikalen Blickwinkel. Die Geschwindigkeit von Gehen und umschaun ist unabhängig von der Framerate.

Springen kann man mit der Leertaste.

Camera

Die Kamera ist eine First-Person-Kamera, die mit der Maus gesteuert wird, so ist es möglich, sich an der Stelle, an der man sich befindet, umzuschauen.

Die Berechnung der View-Matrix wurde von² übernommen.

Textures

Die Texturen werden aus externen Dateien geladen und auf die Objekte angewendet. Die UV-Koordinaten für die einzelnen Objekte werden ebenfalls aus den Model-Dateien geladen.

Moving Objects

Die Objekte in der Welt, die sich bewegen, sind die zwei Autos, die auf der Straße fahren.

Die Geschwindigkeit der Autos ist unabhängig von der Framerate.

Beleuchtung

Die einzige Beleuchtungsquelle parallel einfallendes Licht (Sonnenlicht), außerdem sind alle Objekte entsprechend ihrer Materialeigenschaften auch mit Umgebungslicht beleuchtet.

Adjustable Parameters

Alle einstellbaren Parameter befinden sich in der ini-Datei im Ordner „assets“. Die Helligkeit kann zusätzlich zur Laufzeit durch Drücken der Taste „B“ erhöht werden.

View-Frustum Culling

View-Frustum culling wurde implementiert, sodass nicht immer alle Objekte gezeichnet werden müssen. Die grundsätzlichen Ideen wurden in ³ gefunden.

Jedes Objekt erhält beim Erstellen eine Bounding Box, die das Objekt komplett einschließt. Es werden nun alle Objekte nicht gezeichnet, bei denen sich alle Eckpunkte der Bounding Box außerhalb von mindestens einer der Grenzflächen des View Frustums befinden. Diese Überprüfung wird im Clip Space gemacht.

Es wird am Bildschirm angezeigt, wie viele Objekte zurzeit gezeichnet werden. Mit F8 kann man das Culling ein- und ausschalten.

HUD

Es wurde ein Heads Up Display implementiert, das in der Lage ist, Text am Bildschirm anzuzeigen. Als Textfarbe ist rot und weiß möglich.

Implementiert wurde das HUD durch einen eigenen Shader, der als Koordinaten direkt Bildschirmkoordinaten übergeben bekommt. Gleichzeitig bekommt er eine Textur übergeben, die alle Buchstaben enthält, die dargestellt werden können. Die UV-Koordinaten werden entsprechend dem anzuzeigenden Buchstaben an den Shader übergeben.

Die Idee dazu stammt von ⁴.

Das HUD kann mit der Taste F7 ausgeschaltet werden.

Die angezeigten Debug-Informationen (FPS und gezeichnete Objekte) können mit F6 ausgeschaltet werden.

Physik-Engine

Es wurde keine fertige Physik-Engine verwendet, sondern das gewünschte Verhalten selbst implementiert.

Die Physik-Engine ist in der Lage, Gravitation zu berechnen (beim Springen wird realistisches Gravitationsverhalten modelliert) und Kollisionen zu erkennen. Dafür werden alle Bereiche, bei denen Kollisionen auftreten können, abgespeichert und überprüft, ob eine Kollision auftritt. Es sind auch Kollisionen mit sich bewegenden Objekten möglich.

Effekte

Lighting – Shadow Maps using PCF

Für die Shadow map wird zuerst jedes Objekt aus Sicht der Lichtquelle in eine Depth Map gerendert. Danach wird im Shader überprüft, ob das zu zeichnende Fragment aus Sicht der Lichtquelle vor oder hinter einem anderen Objekt liegt. Liegt es hinter einem anderen, liegt es im Schatten und muss dunkler gezeichnet werden.

Die Idee dazu stammt von ⁵.

Animation – Hierarchical Animation

Animation wurde für sich drehende Räder der Autos implementiert. Dafür sind die Räder des Autos in eigenen Meshes abgespeichert. Die Räder werden dann entsprechend der Bewegung des Autos schneller oder langsamer rotiert.

Texturing – Environment Map

Für das Environment Mapping wird eine Cube Map generiert und ganz im Hintergrund gezeichnet. Das wird von einem eigenen Shader übernommen. Die gleiche Cube Map wird auch an den normalen Shader übergeben und anhand des reflektierten Strahls von der Kamera zum Fragment die gespiegelte Umgebung auf das Objekt gezeichnet.

Ideen hierzu stammen von ⁶

Gespiegelt wird die Umgebung im Wasser und in den Scheiben (vorne und hinten) der Autos.

Post Processing – Lens Flares

Lens Flares wurden implementiert indem zuerst berechnet wird, ob sich die Lichtquelle (Sonne) im Sichtbereich befindet, wenn sie sich innerhalb befindet, werden ihre Koordinaten am Bildschirm berechnet und der Tiefenbuffer an dieser Stelle ausgelesen; damit kann kontrolliert werden, ob die Sonne von etwas anderem verdeckt ist.

Ist sie nicht verdeckt, werden von einem eigenen Shader Texturen auf den Bildschirm gezeichnet, die die Lens Flares darstellen. Es werden unterschiedliche Texturen in unterschiedlichen Transparenzen abhängig vom Abstand der Sonne zum Mittelpunkt des Bildschirms gezeichnet.

Libraries

Als externe Libraries wurde – abgesehen von den Libraries des ECG-Frameworks – bis jetzt nur der Model-Importer ASSIMP verwendet. ¹

Anleitung

Man startet immer an der gleichen Stelle in der Welt. Nun muss man sich durch die Welt bewegen und das Ladekabel suchen, welches sich an verschiedenen Stellen in der Welt befinden kann. Ob das Kabel bereits gefunden wurde, wird in der rechten unteren Ecke des Bildschirms angezeigt. Eingesammelt wird das Kabel indem man sich nahe genug an das Kabel heran bewegt.

Hat man das Ladekabel eingesammelt, muss man zur Tür des Hauses gehen und hat dann gewonnen.

Um sich durch die Welt zu bewegen muss man Wasser überqueren, was nur auf den Plattformen möglich ist, man muss also von Plattform zu Plattform hüpfen.

Hat man das Wasser überquert, muss man die Straße überqueren, ohne von einem Auto berührt zu werden.

Der verbleibende Akkustand wird in der linken unteren Ecke des Bildschirms angezeigt. Man hat insgesamt 3 Minuten Zeit bevor der Akku leer ist; in dieser Zeit muss das Spiel gewonnen werden.

Berührt man das Wasser oder ein Auto, bewegt man sich aus der Welt hinaus, oder ist der Akkustand bei 0%, hat man verloren.

¹ <http://www.assimp.org/>

² <https://www.3dgep.com/understanding-the-view-matrix/>

³ <http://www.lighthouse3d.com/tutorials/view-frustum-culling/>

⁴ <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-11-2d-text/>

⁵ <http://www.opengl-tutorial.org/es/intermediate-tutorials/tutorial-16-shadow-mapping/>

⁶ <https://learnopengl.com/Advanced-OpenGL/Cubemaps>