

Dead Segwace 2

Background

Nach dem der Held aus „Dead Space“ das tentakelige Riesenmonster getötet hatte, machte er sich mit seinem Raumschiff auf den Weg nach Hause. Leider ging ihm auf halber Strecke der Treibstoff aus. Da er nun mit großer Wahrscheinlichkeit den Rest seines Lebens mitten im Weltraum festsitzen würde, tat er das einzig Sinnvolle, was man in so einer Situation tun könnte: Er baute sich eine Segway-Rennstrecke quer durchs All. Von da an verbrachte er jede freie Minute damit, seine Bestzeit zu verbessern. Und er lebte glücklich und zufrieden, bis ihm die Nahrungsreserven ausgingen.

Controls

W	Beschleunigen
S	Bremsen
A	Nach links lenken
D	Nach rechts lenken
Space	Gravity-Twist
R	restart
F2	Framerate in Konsole ausgeben
F3	Wireframe Mode on/off (ACHTUNG: Bloom und Motion Blur werden dabei auch abgeschaltet!)
F4	Texture Sampling Modes
F5	Mipmapping Modes
F8	Einschalten unseres misslungenen Shadow-Mapping
F9	Transparenz on/off
F10	Bloom+Motionblur on/off
F11	Musik on/off
F12	Free-Camera-Mode on/off

External Libraries

- *Coldet*: collision detection <http://sourceforge.net/projects/coldet/>
- *DevIL*: image loading <http://openil.sourceforge.net/>
- *GLEW*: OpenGL library <http://glew.sourceforge.net/>
- *GLFW*: Window creation, input Handling, .. <http://www.glfw.org/>
- *GLM*: Mathematics <http://glm.g-truc.net/>
- *irrKlang*: Audio <http://www.ambiera.com/irrklang/>

Modeling Tools

- Autodesk Maya

Features & Implementation

Geometry Handling

Verwendetes Format

.OBJ; Abgesehen von der High-Polygon-Raumstation, die an einem Teil der Strecke auftaucht, ist der komplette Content selbst in Maya erstellt worden.

Import

Selbst geschriebener Importer. Importiert .obj- und zugehörige .mtl-Datei. Konnte anfangs Ambient-, Diffuse- und Specular-Farbe + Specular Koeffizient importieren; Da wir aber in unserem Programm nur den mitgelieferten Diffuse-Farbwert verwenden, wurde der Rest aus Performancegründen wieder „abgespeckt“.

Wichtiges Feature: Man kann ein Modell auch pro Vertex zufällig einfärben! (Kommt beim „New Record“-Schild zum Einsatz)

VBOs & VAOs

Culling

Ab einer bestimmten Entfernung von der Kamera werden im Shader keine Licht- Farb- und sonstigen Berechnungen mehr durchgeführt.

Graphics

Blinn Phong Shading

Texturing

Texturen (und Normal Maps) werden mit DevIL importiert. Eine Textur kann für mehrere In-Game-Objects wiederverwendet werden; allerdings kann pro Mesh nur eine Textur verwendet werden.

Mipmapping wurde implementiert, ist aber standardmäßig ausgeschaltet, weil wir finden, dass das besser aussieht. Die Mipmapping-Modi können aber mit F5 gewechselt werden, sowie die Texture-Sampling-Modi mit F4.

Normal Mapping

Werden genauso erzeugt wie Texturen, nur später als Normalmaps verwendet.

Transparenz

Einfache Implementierung: Schwarze Stellen (0,0,0) auf der Textur werden vom Fragment Shader als transparent interpretiert und der Alpha-Wert wird auf 1 gesetzt.

Licht

Es gibt 2 Lichtquellen im Spiel: Eine sitzt auf dem großen blauen Stern auf der Skybox und wandert mit der Skybox mit, die andere fährt vor dem Segway her (als „Scheinwerfer“).

Bloom

Wurde implementiert wie im CG2 Repititorium und für bestmögliche Bildqualität fine-tuned.

Radial Blur

Fungiert als eine Art Motion-Blur; Die Intensität des Effekts hängt ab von der Geschwindigkeit des Fahrers und vermittelt den Eindruck von atemberaubender Geschwindigkeit.

Gameplay

„Object Groups“

Die In-Game-Objects sind zu dynamisch erstellten „Objectgroups“ zusammengefasst, die verschiedene Eigenschaften haben (z.B. „segwayman“ enthält alle Teile der animierten Spielerfigur, „floor“ sind Teile auf denen man fahren kann, „obstacle“ sind Teile gegen die man nicht fahren darf,...).

Hierarchical Animation

Die Räder drehen sich und der Fahrer legt sich samt der Lenkstange des Segways in die Kurve.

„GUI“, Zeitmessung

Die aktuelle Fahrzeit wird am Rücken des Fahrers angezeigt (Vorbild hierfür war der Lebensbalken aus „Dead Space“ am Rücken des Protagonisten). Leider sieht man die Ziffern durch den Bloom Effekt hin und wieder nicht so gut wie man sollte.

Realisiert wurde das durch sechs 10-eckige Räder, auf denen rundherum in Form einer Textur die Ziffern von 1 bis 10 aufgebracht sind. Diese Räder sind im Rücken des Fahrers eingelassen und werden, abhängig von der glfw-Zeitmessung, um den richtigen Winkel gedreht, sodass sich die aktuelle Zeit in Minuten, Sekunden und Hunderstel ergibt. Nach Beendigung eines Rennens bleibt die Anzeige stehen, man sieht also gleich wie schnell man war.

Man kann sich dann auch in der Konsole die gefahrene Zeit sowie die Rundenzeit ansehen. Sollte ein neuer Rekord aufgestellt werden, so wird die entsprechende Zeit in die Datei „record.txt“ gespeichert.

Weiters gibt es Blöcke, auf denen „Lap 2“ bzw. „New Record“ steht. Diese werden ständig hinter der Kamera „nachgezogen“ und im Bedarfsfall vor die Kamera bewegt.

Die Einbindung einer externen GUI-Library erschien für unsere Zwecke nicht rentabel.

„Weiche“ Kameraführung

Die Kamera fährt stets hinter dem Fahrer her, aber „gibt nach“ wenn sich der Spieler zur Seite dreht. Dadurch entsteht ein sehr natürliches Spielgefühl.

Geschwindigkeitsabhängiges Ansprechen der Lenkung

Je schneller man fährt, desto unflexibler die Lenkung

Fahren auf der Strecke, fallen, gravity Twist

Grundsätzlich wird stets überprüft, ob die Räder des Segways mit der Strecke kollidieren (mit der „Coldet“-Library). Wenn dem so ist, fährt man geradeaus weiter. Wenn nicht, und wenn auch kein Gravity-Twist am Laufen ist, fällt man nach „unten“ (wo das ist, kommt auf die momentane Wschwerkraftslage an). Erreicht der Spieler im Zuge des Fallens einen gewissen y-Wert, stirbt er. Man kann übrigens keinen Gravity-Twist starten, wenn man gerade fällt, sonst wäre das Spiel ein bisschen witzlos.

Explosionen

Fährt man gegen eine Kiste mit der Aufschrift „Explosive“, so explodiert diese (Überraschung!) und man stirbt. Die Explosion ist durch ein Polygon-Quadrat realisiert, auf dem eine Explosions-Textur aufgebracht ist. Dieses Quadrat führt der Fahrer irgendwo im Brustbereich mit sich herum und beim Zusammenstoß mit einer Kiste wird es, mit quadratisch abfallender Geschwindigkeit, zu einer passablen Explosion skaliert. Dies ist übrigens die Einzige Stelle wo unsere Transparenz zum Einsatz kommt (damit man das Quadrat nicht sieht).

Skybox

Eine Skybox, die mit der Kamera mitfährt, vermittelt den Eindruck vom unendlichen Weltraum.

Sound

Motorengeräusche, deren Lautstärke und Abspielgeschwindigkeit von der Geschwindigkeit des Segways Abhängen; Bremsen, Explosion, Wilhelm Scream; stillvoller Soundtrack.

Warum es keine Schatten gibt

Normalerweise gibt es im Weltraum keine Schwerkraft und keine Schallausbreitung. In unserem Weltraum schon. Als ausgleichende Gerechtigkeit gibt es in unserem Weltraum dafür keine Schatten. Damit der echte Weltraum nicht neidisch wird, quasi.