



Computergrafik Projekt

03.09.2018

Rothert, Christopher (723863)

Werner, Alexandra (723918)

Kurzbeschreibung

Der Spieler muss versuchen, entgegenkommenden LKWs auszuweichen. Dabei wird eine leicht schwenkende Kamera aus der dritten Person eingesetzt, die dem Spieler einen guten Überblick über die Spielsituation geben soll. Während des Spiels tauchen Münzen auf, die der Spieler einsammeln kann.

Ziel des Spiels

Das Ziel des Spiels ist es, solange wie möglich entgegenkommenden LKWs auszuweichen. Ein Timer misst dabei die Dauer des aktuellen Spiels. Die längste Zeit wird festgehalten und zusammen mit der Menge der eingesammelten Münzen zwischengespeichert.

Steuerung

Der Spieler kann mit den Tasten A und D die Bewegungen der Spielfigur steuern. Zusätzlich hat der Spieler die Möglichkeit, seine Geschwindigkeit kurzzeitig zu erhöhen, indem er die linke Shift-Taste drückt. Nach dem Benutzen der Shift-Taste, ist diese Funktion vorübergehend deaktiviert, damit sie nicht missbraucht werden kann.

Umsetzung

Programmierung

Gameplay

Die Steuerung des Spielers wurde so programmiert, dass sich die Spielfigur über die Tasten "A" und "D" nach links und rechts bewegen kann. Außerdem ist es dem Spieler mit der linken Shift-Taste möglich den LKWs mit einer erhöhten Geschwindigkeit auszuweichen. Diese Animation findet über eine Matrix-Transformation statt.

Der Spieler kann außerdem Münzen einsammeln, die über dem Spielfeld schweben. Zur Veranschaulichung werden dabei Münzen neben dem Spielfeld gestapelt.

Um zu verhindern, dass der Spieler sich außerhalb des Spielfeldes bewegt, wird die Spielfigur auf der X-Achse der 3D-Umgebung eingeschränkt. Somit ist es dem Spieler nicht möglich, die Situation auszunutzen, allen LKWs ohne Bewegung auszuweichen.

LKW

Die Lastkraftwagen bewegen sich vom hinteren Spielbereich auf die Spielfigur zu. Die Farbe des Lichtes aus ihren Scheinwerfern wird zufällig generiert.

Grafik

3D-Modelle

Zur Erstellung der 3D-Modelle sowie UV-Maps wurden zwei Programme verwendet, das aus dem Modul "3D-Modelling und Animation" bekannte Cinema4D, sowie auch Blender. Für die Erstellung von Normal Maps wurde auf Photoshop zurückgegriffen.

Für den Spieler steht außerdem noch eine alternative Spielfigur im assets-Ordner zur Verfügung (player.dae und player2.dae)

Texturen

Die Texturen der Modelle wurden sowohl über die Google Bildersuche als auch mit Cinema4D erstellt. Teilweise wurde auf bereits vorgefertigte Cinema4D Texturen aus der Studenten Version zurückgegriffen. Bei allen Texturen wurde auf das Creative Commons License Recht geachtet.

Licht

Zusätzliche Lichtquellen wurden im Level verteilt, um die Atmosphäre zu verbessern

Audio

Das Spiel wurde mit Musik hinterlegt. Diese wurde mit dem Programm "GarageBand" erstellt und durch die "irrKlang" API in dem Projekt eingefügt. Es wurden insgesamt zwei Tracks selbst erstellt, die sich im assets-Ordner befinden. (bgm_01.mp3 und bgm_02.mp3)

Quelle: <https://www.ambiera.com/irrklang/index.html>

Weitere Soundeffekte wurden von folgenden Seiten heruntergeladen:

- <https://soundbible.com>
- <https://freesound.org>

Einschränkungen

Lichtquellen

Da die Anzahl der Lichtquellen limitiert ist, mussten Abschnitte des Levels, die dem Spieler nicht gezeigt werden, ohne Lichtquellen versehen werden, damit das Spielerlebnis nicht darunter leidet.

Besonderheiten

Besonders hervorzuheben sind folgende Funktionen:

- Dynamischer Schwierigkeitsgrad: LKWs werden auf Dauer schneller, anpassbar über Variablen im Konstruktor von Application.cpp.
- LKWs und Münzen werden in einem Objektpool gespeichert und "recycled", damit die Anwendung während der Laufzeit nicht unerwartete Framedrops/Frameskips verursacht. Methode createBaseScene() in Application.cpp.
- Lichter der LKWs sind zufällig farblich angepasst, wenn sie das erste Mal geladen werden oder wenn sie zurückgesetzt werden, um das Spielerlebnis zu verbessern. Methode loadModels() und reset() in Truck.cpp.
- Kleine Animation der Spielfigur. Methode animate(float dtime) in Player.cpp
- Kollisionsabfrage zwischen Spieler und LKWs/Münzen. Methode checkCollision(const Model* other) in Player.cpp
- Für eine umfangreichere Kollisionsabfrage wurde die Header-Datei Model.h um eine Funktionalität erweitert: Die Methoden getModelType, setModelType und die Variable type ermöglichen die Beschreibung des Modeltyps

TODO

Aus zeitlichen Gründen konnten nicht alle von uns gewünschten Funktionen umgesetzt werden. Dazu gehören:

- GUI (Timer, Münzstand)
- (Schattenberechnung aus Praktikum)
- (Physics)
- (Stationäre und/oder bewegliche Gegner, die bekämpft werden können)