

Abgabe 2 - Dokumentation

Sled Race

186.831-2014S - Computergraphik UE

23.07.2014

*Bernhard Rainer
David Peherstorfer*

*0828592
0929021*

*b_rainer@gmx.net
david.peherstorfer@live.de*

Beschreibung

Sled Race ist ein 3D Schlittenrennspiel, bei dem der Spieler einen Jungen auf einer Rodel steuert. Er muss Hindernissen ausweichen, Punkte sammeln und zeitgerecht ins Ziel kommen. Erreicht der Spieler einen Checkpoint, wird ihm Zeit gutgeschrieben.

Die Kamera folgt dem Charakter in einer 3rd Person View.

Steuerung

Mit den Pfeiltasten wird der Schlitten gesteuert. Hierbei beschleunigt der Schlitten bei Pfeil-Rauf und bremst bei Pfeil-Runter. Pfeil-Links und Pfeil-Rechts drehen den Schlitten jeweils in die gewünschte Richtung. Zusätzlich kann zu Entwicklungszwecken die Kamera mit einem Druck auf die F-Taste zwischen 3rd Person und 1st Person wechseln.

Pfeiltaste Rauf	Beschleunigen
Pfeiltaste Runter	Bremsen
Pfeiltaste Links	Links
Pfeiltaste Rechts	Rechts

Implementierung

Beleuchtung

Die Szene wird von einer globalen Lichtquelle, ähnlich der Sonne, beleuchtet. Hierbei wurde ein Lambert-Gouraud-Beleuchtungsmodell implementiert. Die Beleuchtung beeinflusst alle Objekte der Szene gleich.

Kamera

Eine Kamera folgt dem Spieler in einer 3rd Person View. Zu Entwicklungszwecken kann der User die Kameraansicht wechseln und sich in der Ego-Perspektive durch die Spielwelt bewegen.

Komplexe Objekte

Der Schlittenfahrer stellt sich einerseits aus einem Torso, mit zwei beweglichen Beinen zusammen, sowie dem Schlitten als Untersatz. Die einsammelbaren Items bestehen ebenfalls aus komplexen Triangle-Meshes. Das Terrain wurde aus einer Height-Map generiert, aber vorab schon trianguliert.

Animierte Objekte

Beim Lenken und Bremsen bewegt der Schlittenfahrer seine Beine auf der gewünschten Seite.

Transparency

Die Partikel-Systeme verblassen nach einer gewissen Zeit immer mehr, um die natürliche Auflösung einer solchen Wolke zu simulieren. Gleichzeitig nutzen verschiedene Interface-Elemente den zusätzlichen Alpha-Kanal um eine schönere Überblendung zu erzielen.

Features

GPU-based Particle System

Sobald der Spieler einen Fuß auf den Boden setzt (Bremsen und Lenken), entsteht eine Schneewolke, die mit einem GPU-basierten Partikelsystem generiert wird.

Motion Blur

Um eine realistische Geschwindigkeit darzustellen, wird die Szene mittels Per-Object-Motion-Blur verschwommen gezeichnet. Motion Blur funktioniert auf AMD und Intel Grafikchips.

Shadow Mapping

Die Szene wird von einem Directional-Light beleuchtet. Die Spielmodelle sollen Schatten werfen. Die Schatten werden zunächst in eine Shadow Map gerendert und danach durch einen Depthvergleich in die finale Szene gerendert. Die Textur wird dabei linear gefiltert und über einen 5x5 Kernel geaveraged.

Verwendete Libraries

GLFW3

Fenster-Handling und InputManagement (<http://www.glfw.org/>)

GLFW

OpenGL Extension Wrangler Library (<http://www.glfw.org/>)

NVidia PhysX 3.3

Collision Detection Library (<http://www.nvidia.de/object/nvidia-physics-de.html>)

Assimp

Model Loading Library (<http://assimp.sourceforge.net/>)

GLM

Mathematik-Library für Vektor- und Matrizenberechnungen (<http://glm.g-truc.net/>)

FreeImage

Library for Image Processing (<http://freeimage.sourceforge.net/>)

Verwendete Tools

3DS-Max

Modelling Programm für 3D Meshes und Texturierung

EarthSculptor

Terraforming-Programm zum erstellen von komplexerem Terrain

Google Chrome

Als Research-Quelle

Verwendete Tutorials

Model Loading:

<http://www.lighthouse3d.com/cg-topics/code-samples/importing-3d-models-with-assimp/>

Per Object Motion Blur:

<http://john-chapman-graphics.blogspot.co.uk/2013/01/per-object-motion-blur.html>

Particle System

<http://www.gamedev.net/topic/606935-gpu-based-particle-system/>

<http://antongerdelan.net/opengl/particles.html>

Shadow Mapping:

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