Downhill Race

Implementierung

Spielmechanik
Das erste Schaf ist frei steuerbar mit WASD, das zweite mit den Pfeiltasten, das dritte mit IJKL und das vierte mit Num-8456. Wenn Gamecontroller angeschlossen sind, werden diese verwendet und die Tastatur für den Spieler ignoriert. Wenn das Schaf die Bounding Box der Startfahne unter Annahme eines Toleranzwertes durchquert startet der Timer. Beim durchqueren der Zielfahne wird dieser Timer angehalten und die vergangene Zeit sowie der erreichte Platz werden in der Konsole ausgegeben.

Effekte

Shadowmapping
Shadowmapping wird für alle normalen Objekte verwendet, nicht für Partikel und Volumen-Rendering.

Referenzen:

Folien zum Shadowmapping von den Zwischenvorträgen (Grundlegender Ablauf, Shadow acne, Peter Panning)

http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/ (Poisson Sampling)

GPU-Partikel
Die Partikel werden für die Staubwolken von den Schafen verwendet.

Referenzen:

http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/

Volumen Rendering
Das Volumen Rendering wird für die Wolle der Schafe verwendet.

Referenzen:

http://http.developer.nvidia.com/GPUGems/gpugems_ch39.html (Grundlegendes Verfahren)

Christian Hafner (Danke! ;-))

http://lgdv.cs.fau.de/External/vollib/ (Volumen-Daten, eher nutzlos für uns)

http://lebarba.com/blog/ (leider raycasting und kein slicing)

http://graphicsrunner.blogspot.com.ar/search/label/Volume%20Ray-Casting (auch raycasting)

Komplexe Objekte
So gut wie alle Objekte sind Modelle im .obj – Format. Einfache Objekte wie Ebenen werden nur fürs Volumen-Rendering, Renderbuffer und die Partikel verwendet.

Animierte Objekte
Das Schaf hat Beine, die sich abhängig von der Geschwindigkeit bewegen.

Die Vogelscheuche dreht sich, wenn das Schaf dagegen rennt.

View Frustum Culling

Es werden axis-aligned bounding boxes mit dem View Frustum geschnitten für das Culling. Die Anzeige der gezeichneten Vertices ist im Fenstertitel. Es ist standardmäßig an und kann mit F8 ein- und ausgeschaltet werden.

Transparenz

Das Volumen-Rendering verwendet transparente Scheiben durchs Volumen. Es ist sichtbar am Schaf und bei den Partikeln. Es kann mit F9 ein- und ausgeschaltet werden.

Experimentieren mit OpenGL

Die aktuellen Daten (Frame-Zeit, FPS, gezeichnete Vertices) werden immer im Fenstertitel angezeigt. Die Nachrichten für den State-Wechsel, z.B. bei F3 (Wireframe) werden in der Konsole angezeigt.

Vertex-Buffer-Objects (VBO)

VBOs werden für jedes Modell verwendet. Die Implementierung steckt in der Klasse „VertexBuffer“ in der Resource.hpp.

Vertex-Array-Objects (VAO)

VAOs werden u.a. für die Partikel verwendet. Die Implementierung steckt in der Klasse „VertexArray“ in der Resource.hpp.

Frame-Buffer-Objects (VAO)

VAOs werden überall verwendet. Die ganze Scene landet in einem FBO als Depthtexture fürs Shadowmapping. Fürs Volumen-Rendering gibt es FBOs für den Light- und Eye-Buffer.

Fürs Split-Screen gibt es FBOs für jede Kamera.

Die Implementierung steckt in der Klasse „FrameBuffer“.

Mip Mapping

Eingeschaltet bei allen normalen Texture. Die Unterschiede sind sehr gering. Am besten sieht man sie an der Bodentextur.

Texture-Sampling-Qualität

Eingeschaltet bei jeder Textur. Der Unterschied ist vor allem bei der Schafwolle deutlich.

Tastenbelegung

F3 – Wireframe

F4 – Textur-Sampling-Qualität

F5 – Mip Mapping-Qualität

F8 – Viewfrustum Culling

F9 – Transparenz

Kamera

Die Kamera passt sich an die Bewegungsrichtung des Schafs an. Da es oben und unten nichts weiter zu sehen gibt, kann die Kamera indirekt durch das Schaf in alle interessanten Richtungen gedreht werden.

Bewegende Objekte
Im Moment gibt es eine Vogelscheuche und das Schaf als einzige bewegte Objekte. Es gibt noch andere Schafe im Multiplayer und diverse Hindernisse auf der Strecke.

Texturen
Es werden verschiedene Texturen auf dem Boden, dem Schaf, der Start- und Zielflagge, den Bäumen angezeigt.

Beleuchtung und Material
Es gibt eine gerichtete Lichtquelle. Alle Objekte haben fast das gleiche Material: Die gleichen Shader mit dem Phong-Modell, underschiedliche Meshes mit Normalen und UV-Mapping.

Steuerung
Das erste Schaf ist frei steuerbar mit WASD, das zweite mit den Pfeiltasten. Die Kräfte, die auf das Schaf wirken sind abhängig von der Kameraposition. Für die Vorwärtsbewegung beispielsweise wird der Vektor von der Kamera zum Schaf (ohne y-Komponente) mit einer Konstanten und der deltaTime multipliziert und als Impuls angewendet.

Tools
Visual Studio 2013

Maya

gDEBugger

Notepad++

Sourcetree (Git-Client)

Features
Supergeile Physik

Split-Screen

Gamepad-Unterstützung

Hintergrundmusik

Bibliotheken
Alle Bibliotheken werden statisch gelinkt. Es sind also keine zusätzlichen DLLs nötig.

Die Grundlagen zu OpenGL, Basic Shading, Modelloader, Textureloader basieren auf folgendem Tutorial: http://www.opengl-tutorial.org/

Kollision und Festkörperphysik sind mit Bullet gemacht: http://bulletphysics.org/wordpress/

GLFW Window and Input Handling: http://www.glfw.org/

GLM OpenGL Mathematics: http://glm.g-truc.net/0.9.6/index.html

GLEW OpengGL Extensions: http://glew.sourceforge.net/

SDL für Musik http://www.libsdl.org/index.php