Fischer Thomas
gs16m022

EZG Endabgabe
FishGL

Steuerung

WASD + Maus	Kamera bewegen
T	Kamerafahrt toggle
R	Kamerafahrt Keyframe änder (max 10)
L	Lichtposition ändern (Toggle)
N+A|D	N gedrückt lassen und A oder D um die “Bumpiness” zu verändern
P	Schatten toggle
1-5+I	AA-Samples 2-32; I um ausgewählte Samplesize zu aktivieren
Q	Ray feuern

Module

Kamerafahrt

Die Kamera wie auch die Keyframes werden als struct aus Position (glm::vec3) und Rotation (glm::quaternion) abgespeichert. Die gesamte Animation wird in einem Animation-Struct abgelegt. Dieses besteht aus einem Keyframe-Array, der Anzahl an Frames als int und der Dauer als float.
Wird die Animation nun abgespielt wird durch alle Keyframes durchiteriert. Die Animation läuft als Loop ab. Für die Interpolation wird glm::catmullRom (Position) und glm::squad (Rotation) verwendet.

Fischer Thomas
gs16m022

Probleme

Leider konnte ich bis heute nicht herausfinden warum die Squad-Interpolation in meiner Implementierung solche “Schwenker” verursacht. Diese Probleme bestehen beim Verwenden von glm::mix oder glm::lerp nicht.
Zusätzlich kommt es bei einer Drehung um mehr als 180° zu einem interessanten Flip der Kamera. Dieser wird verursacht da Quaternions nicht mehr als -180° bis +180° abspeichern können und bei meiner Implementierung die Gesamtrotation pro Keyframe abgespeichert wird. Aus Zeitmangel konnte dieser Fehler jedoch nicht behoben werden.

Schatten (Directional Light)

Das Licht wird in einem struct mit Position (glm::vec3) und Farbe (glm::vec3) abgespeichert. Zusätzlich werden 2 unterschiedliche Shaderprogramme verwendet. Einmal um die Depthmap zu bekommen und einmal der normale Shader um die Szene auch normal rendern zu können.

Normalmapping

Um eine Normalmap richtig verwenden zu können müssen die Vertices über Tangents verfügen. Der TinyObjLoader kann diese nicht selber berechnen, daher müssen nach dem Laden der Objekte alle Tangents einzeln berechnet und eingefügt werden (siehe FishGL::calcTangents). Zusätzlich mussten noch kleinere Änderungen am Shader durchgeführt werden um besagte Tangents darin zur Verfügung zu haben.

Anti-Aliasing

Um Anti-Aliasing realisieren zu können musste die komplette Szene anstatt direkt zuerst in einen Framebuffer gerendert werden. Dieser wird je nach ausgewählter Samplesize neu generiert (siehe FishGL::i_generateNewFrameBuffer und FishGL::i_generateMultiSampleTexture).

KD-Tree und Rays

Der kdTree wird einmal, direkt nach dem Laden des obj-Files, generiert. Für die Schnittachse wird zyklisch x,y und z verwendet. Abgebrochen wird sobald in eine Node nur mehr 12 Triangles (genau ein Würfel) übergeben werden.

Fischer Thomas
gs16m022

Probleme

Wird für den Raycast der kdTree verwendet werden nur sehr wenige Intersections erkannt. Derzeit wird bei jedem Raycast über alle Triangles iteriert. Aufgrund von akutem Zeitmangel konnte dieses Problem noch nicht behoben werden.