Gesamtabgabe

• Die Projekte 1-5 sind bei dieser Abgabe in einem Programm mit voller Funktionalität abzugeben

• Zusätzlich ist eine schriftliche Ausarbeitung als pdf abzugeben

• Kurz halten (max. 2-3 Seiten)

• Was wurde gemacht? Wie wurden die Algorithmen umgesetzt?

Wo gab es Probleme und wie wurden diese gelöst?

• Video ist nicht erforderlich, da eine mündliche Prüfung darüber stattfindet

• Zoom Link zur Abgabe und Prüfung gibt es rechtzeitig im Moodle Kurs

Abgabe

- Sources and Release auf Github
  https://github.com/incredibleLeitman/TrackingShot

- Release 1.04 für Uebung4
  https://github.com/incredibleLeitman/TrackingShot/releases/tag/1.04

- Programm und Sourcecode Präsentation auf DropBox
  https://www.dropbox.com/sh/bi6lylyfb8hyu4k/AABhwqHoEqe2dTBeu0CwNuXIa

Umsetzung

✓ Modern OpenGL3 mit glew und glfw library, stb_image header für texture handling

✓ Catmull spline – Anlegen von control points im Kreis um das Zentrum mit Ausrichtung zum Folgepunkt. Umschalten des Kameramodus zwischen automatischer Bewegung und manueller Steuerung, in welcher man weitere Punkte hinzufügen kann, mittels key callback. Es werden zyklisch vom aktuellen Wegpunkt die nächsten Punkte als Stützpunkte herangezogen, zwischen denen mit catmullSpline interpoliert wird, zu denen sich die Kamera bewegt. Mittels qml::squad wird die slerp Funktion der Ausrichtung zwischen dem aktuellen und nächsten Punkte evaluiert.

✓ Shadows – Erstellung eines shaders für die erstellte Depthmap. Rendervorgang wird zusätzlich noch einmal pro Lichtquelle extra berechnet: Tiefenwerte werden in einer Textur abgespeichert, welche dann im eigentlichen Renderzyklus herangezogen wird. Phänomen „shadow acne“ gelöst durch die Verwendung eines kleinen bias.

- ✓ Normal mapping – Implemenation eines TexturManagers mittels stb_image. Erweiterung der Shader für die Verwendung einer Textur für die eigentlichen Pixel Farben sowie für die normalmaps. Berechnung der normals passiert noch im shader, was relativ ineffizient ist, jedoch war das aufgrund der bereits etablierten Datenstrukturen die einfachste Möglichkeit. Das könnte man noch schöner machen, indem man Vertexbuffer verwendet. Diese werden um einen bumpiness-Faktor modifiziert mit welchem sich die Stärke der Auswirkung einstellen lässt.

- ✓ Anti aliasing – Einschalten des multisamplings möglich über die glfw boardmittel: kann einfach vor dem Erstellen eines windows angegeben werden. Der Modus kann bei der Implementierung über die Funktionstasten F1 – F12 gesetzt werden. Leider ist bei dieser simplen Lösung keine Modifikation des Modus zur Laufzeit möglich, weshalb bei einer Änderung das Fenster neu erstellt an der Stelle des aktuellen geöffnet wird. Die Option, das AA selbst zu implementieren war aufgrund von Zeitmangel nicht möglich.

- ✓ kd-Tree und Intersection – wurde aus Zeitgründen nicht gemacht

- ✓ Allgemein – Prinzipiell wurden die theoretischen Konzepte bereits in den Vorlesungen gut durchbesprochen und erklärt, unter Zuhilfenahme der empfohlenen Literatur konnte das Wissen noch vertieft werden. Die größten Probleme hatte ich zu Beginn mit der Auswahl der Technologie: Da GLUT prinzipiell als deprecated angegeben wurde und nicht mehr empfohlen wird zu verwenden, habe ich mir eine Reihe von Alternativen angesehen und das Projekt in drei unterschiedlichen Variationen realisiert. Letztendlich hat das viel Zeit auf Kosten der Qualität beansprucht, die ich im Laufe des Semesters nicht mehr nachholen konnte. Die Entscheidung auf modernes OpenGL zu setzen hat sich im Nachhinein betrachtet jedoch bewährt, da viele der Aufgaben schön im shader Teil der Implementierung gelöst werden konnten. Auch war die Verfügbarkeit von modernen OpenGL Tutorials viel größer und die Dokumentation zu häufigen Fehlern viel verbreiteter.