### Mehr Texturierung

# Aufgabe 1

In der Datei perspective. asy auf GRIPS ist eine vollständige Render-Pipeline in Asymptote implementiert. Vieles davon kennen Sie schon bzw. haben es sogar schon in vorigen Übungsaufgaben implementiert.

Im ersten Teil (bis Zeile 230) ist deshalb primär die Klasse vertex\_attributes relevant. Die Eckpunkte eines Dreiecks werden mit Hilfe dieser Struktur abgebildet, sie enthällt z.B. die Position des Eckpunkts, aber auch alle Felder die für perspektivisch korrekte Interpolation nötig sind. In diesem Teil ist es *nicht* nötig Code anzupassen.

Die Dreieckspositionen und -texturkoordinaten werden im dritten Teil (ab Zeile 278) je nach Konfiguration via mode so definiert, dass die zwei Dreiecke in einer zur Blickrichtung orthogonalen Ebene liegen, bzw. die Ebene mehr oder weniger schräg steht. Passen Sie hier *nur* mode an um zwischen den verschiedenen Varianten der Ebene zu wählen.

Im zweiten Teil (Zeilen 230 – 278) wird bisher nicht perspektivisch korrekt interpoliert. Analog zur Aufteilung in der Vorlesung gibt es einen programmierbaren Teil, den Vertex Shader (vertex\_shader) und den Fragment Shader (fragment\_shader), sowie einen nicht programmierbaren (in OpenGL bspw. in Hardware angenommenen) Teil (fixed\_function\_vertex und fixed\_function\_fragment).

Implementieren Sie die perspektivisch korrekte Interpolation ausschließlich im fixed function Teil.

## Aufgabe 2

Im Tarball cg-2021-a03.tar.gz finden Sie eine erweiterte Variante des Codes der letzten Aufgabe zur Texturierung der Sie in dieser Übung weitere, fortgeschrittene Texturierungsmethoden hinzufügen sollen.

#### Überblick.

Im Vergleich zur letzten Vorlage hat sich nur sehr wenig geändert, es gibt neue Einstellungsmöglichkeiten für die Teilaufgaben und entsprechend neue Shader Dateien. C++-seitig ist alles vorbereitet, so dass Sie nur Shader-Code anpassen müssen.

Wenn Sie im Drop-Down Menü 'Masked' auswählen wird zum Rendern der Szenengeometrie der Shader shaders/mask. {vert,frag} verwendet. Vervollständigen Sie diesen, so dass, falls für das gerade gezeichnete Drawelement eine Maskierungstextur definiert ist, die entsprechende Maske auf die Fragmente angewendet wird (recherchieren Sie den GLSL Befehl discard).

Wie Sie aus der VL wissen ist die Maskierungsinformation erstmal binär. Welchen Einfluss haben die verschiedenen Textur-Sampling Methoden darauf, welchen Wert wollen Sie im Shader als Referenzwert verwenden?

Außerhalb des maskierten Bereichs sind die Farbtexturen (siehe Textur-Dateien oder Darstellung in der VL) nicht schwarz, sondern im selben Ton gehalten wie der nicht ausmaskierte Bereich. Warum ist das sinnvoll? Vergleichsbilder finden Sie in den Vorlesungsfolien.

#### Happy Hacking:)

Kai Selgrad 1 OTH Regensburg