

## Real Time Rendering Aufgabe 2: Terrain Fly-Over

**Abgabe: siehe Moodle!**

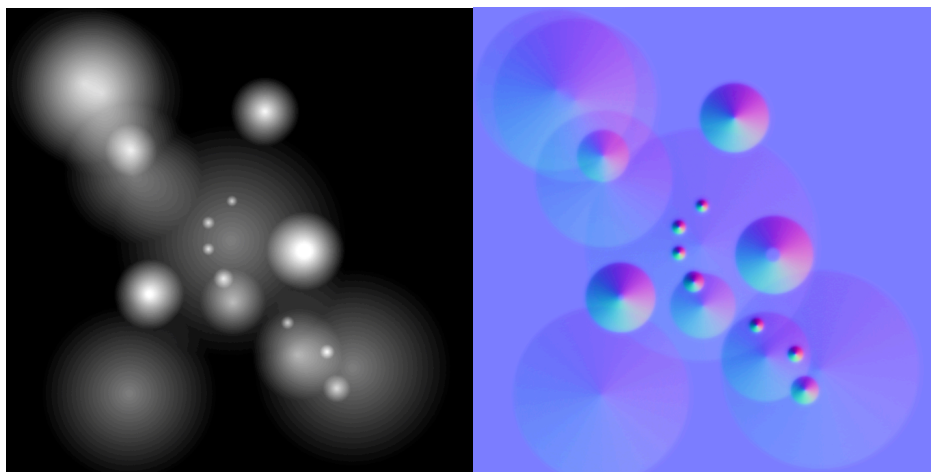
Demonstrieren Sie in der Übung nach der Abgabe Ihre Lösung. Alle Gruppenmitglieder müssen dazu anwesend sein. Ausschlaggebend ist neben der eigentlichen Lösung vor allem die Qualität Ihrer Erklärungen.

Bewertung: Werden die Teilaufgaben 3.1-3.3 gelöst, wird die Aufgabe maximal mit *gut* bewertet. Ein *sehr gut* ist nur bei Lösung aller Teilaufgaben möglich. Verspätete Abgaben werden mit 2/3 Notenabzug pro angefangener Woche Verspätung belegt.

### Aufgabe 2.1: Terrain

Implementieren Sie ein Terrainmodell, indem Sie eine fein tesselierte, ebene Geometrie mit Hilfe von Displacement- und Bump-Mapping als erhabene Landschaft darstellen. Sollten Sie keine geeignete Höhenkarte finden, so können Sie diese auch als Grauwertbild selbst erstellen, z.B. mit Gimp. Aus einer Höhenkarte lassen sich z.B. mit CrazyBump die notwendigen Höhenfelder und Bump Maps erzeugen.

Um eine ebene Geometrie zu erstellen, können Sie sich der Klasse `Rect` aus dem Textur-Beispiel bedienen. Die `texture_demos` Beispiel-App stellt einen guten Startpunkt für die Lösung dieser Aufgabe dar.



Verwenden Sie verschiedene Texturen, um die Oberflächenbeschaffenheit des Terrains in Abhängigkeit von Höhe und Steigung des Geländes zu modellieren. Durch die Mischung von geeigneten Texturen wie Gras, Sand und Geröll, lässt sich ein interessantes und abwechslungsreiches Aussehen erzeugen. Hier zählen Kreativität und Realismus.

## Aufgabe 2.2: Steuerung und die Unendliche Weite

Implementieren Sie eine Steuerung für die Kameraposition, die es dem Benutzer erlaubt den Flug intuitiv und ohne Ruckeln über das Terrain zu steuern. Mindestens soll es möglich sein, Geschwindigkeit, Flughöhe und Flugrichtung zu beeinflussen.

Beachten Sie, dass Flugzeuge typischerweise nicht abrupt in der Luft stehen bleiben. Nützlich zur Entwicklung eines Flugmodells sind die physikalischen Betrachtungen der gleichförmigen, gleichmäßig beschleunigten Bewegung. Die Simulation von wirklich realistischem Flugverhalten ist nicht notwendig, aber möglich, wenn auch verhältnismäßig aufwendig (siehe z.B. <sup>1</sup>).

Beachten Sie, dass für eine „ruckelfreie“ Navigation nicht die Wiederholrate der Tastatur (wie oft pro Sekunde feuert eine dauerhaft heruntergedrückte Taste) zählt, sondern die Framerate. Merken Sie sich den Zustand der relevanten Tasten, und fragen Sie vor jedem `draw()` deren aktuelle Zustand ab.

Entwickeln Sie eine Strategie, mit der sich der Eindruck einer endlosen Landschaft erzeugen lässt, ohne endlos viel Geometrie zu definieren. Die periodische Wiederholung der Landschaft ist zulässig. Implementieren Sie den Flug über dieses endlose Terrain.

Tipp: es lohnt sich, Zeit zu investieren, um auf eine möglichst einfache Lösung zu kommen. Denken Sie an die Wolken aus dem Texturbeispiel...

## Aufgabe 2.3: The Sky is the Limit / Hinter'm Horizont....

Implementieren Sie zur Darstellung des Himmels über Ihrem Terrain eine Skybox. Implementieren Sie dazu *Cube Mapping*. Bei der Darstellung der Skybox benötigen Sie Kontrolle über das Backface-Culling und die Verwendung des Z-Buffers.

Recherchieren Sie Techniken zur Darstellung von (sehr einfachen!) Nebeleffekten. Verbergen Sie den Horizont Ihres Terrains in dichter werdendem Bodennebel, so dass das Ende ihres Landschaftsmodells nicht wahrnehmbar ist. Achten Sie darauf, dass der Nebeleffekt nur in Bodennähe auftritt, damit die Skybox nicht vollständig im Nebel versinkt.

---

<sup>1</sup> <http://www.mtc.edu.eg/ASAT13/pdf/CT31.pdf>