Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Heinrich Hußmann
Henri Palleis

Übungsblatt 5: Modellierung

Abgabe:
Dieses Übungsblatt ist einzeln zu lösen. Die Lösung ist bis **Montag, 04.Juni 2012, 12:00 Uhr s.t.** über UniWorx (https://uniworx.ifi.lmu.de/) abzugeben.
Für Textantworten werden nur die Formate PDF, Plain-Text (UTF-8) akzeptiert. Benennen Sie die Dateien nach dem Schema <Übungsblatt>-<Aufgabe>.<extension>, d.h. die Lösung der ersten Aufgabe geben Sie in einer Datei 1-1.txt oder 1-1.pdf ab. Packen Sie alle Dateien in eine ZIP-Datei und laden Sie diese bei UniWorx hoch. Wenn Sie Formatierungsvorgaben nicht einhalten, können ihre Abgaben nicht korrigiert werden.

Aufgabe 1: Ebenengleichung

Gegeben sei ein Tetraeder, der sich aus folgenden Dreiecken zusammensetzt:

- Dreieck 1: p1 = (0, 0, 0), p2 = (2, 0, -2), p3 = (2, 2, 0)
- Dreieck 2: p1 = (2, 0, -2), p2 = (0, 2, -2), p3 = (2, 2, 0)
- Dreieck 3: p1 = (0, 2, -2), p2 = (0, 0, 0), p3 = (2, 2, 0)
- Dreieck 4: p1 = (0, 0, 0), p2 = (0, 2, -2), p3 = (2, 0, -2)

Geben Sie für alle Seiten des Tetraeders die Ebenengleichung in der aus der Vorlesung bekannten Form an.

Aufgabe 2: Heightmap Terrain

Ziel dieser Aufgabe ist es, eine Landschaft zu erschaffen. Ein verbreiteter Ansatz hierfür ist die Verwendung einer Heightmap (siehe Abbildung 1a), wobei es sich um eine Bilddatei handelt, deren Pixelwerte dazu verwendet werden, um die Höhe der Landschaft zu bestimmen.

i. Welche Primitive in OpenGL eignen sich, um ein Polygon-Mesh zu zeichnen, das aus gleichförmigen Dreiecken besteht?

ii. Schreiben Sie ein JOGL-Programm, das ein Raster von 100*100 Punkten erzeugt und verbinden Sie die Punkte so, dass dabei eine ebene Fläche (XZ-Ebene) entsteht, die in lauter gleichförmige Dreiecke aufgeteilt ist (siehe beispielhafter Ausschnitt in Abbildung 1b).
Tipp: Mit `gl.glPolygonMode(GL2.GL_FRONT_AND_BACK,GL2.GL_LINE);` können Sie in den sog. Wireframe-Modus von OpenGL schalten.

iii. Verwenden Sie die Datei *ImageLoader.java* um die Datei *heigthmap.png* (100px*100px) in ihr Programm einzulesen und die Grauwerte jedes Pixels in einem Array zu speichern (Die Dateien sind auf der Webseite zur Vorlesung verlinkt):

```
int[] pixels =
```

```
ImageLoader.loadImage("http://www.medien.ifi.lmu.de/lehre/ss12/cg1/uebung/heightmap.png");
```

iv. Verwenden Sie diese Werte, um den einzelnen Knoten der Ebene einen Wert für deren Höhe zuzuweisen. Zusätzlich soll die Farbe der Knoten von diesem Wert abhängen.

v. Wählen Sie eine geeignete Projektion und platzieren sie die Kamera. Abbildung 2 zeigt eine beispielhafte Ansicht der fertigen Szene.

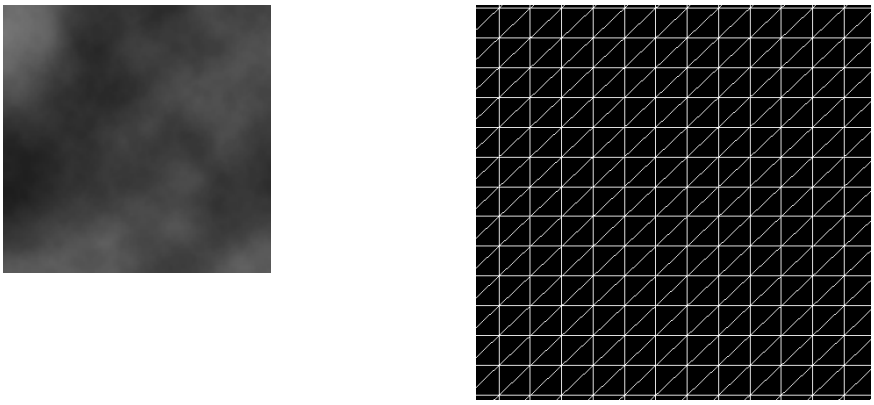

Abbildung 1 a) Eine Heightmap (100px*100px) b) Ein Auschnitt aus dem Polygon-Mesh

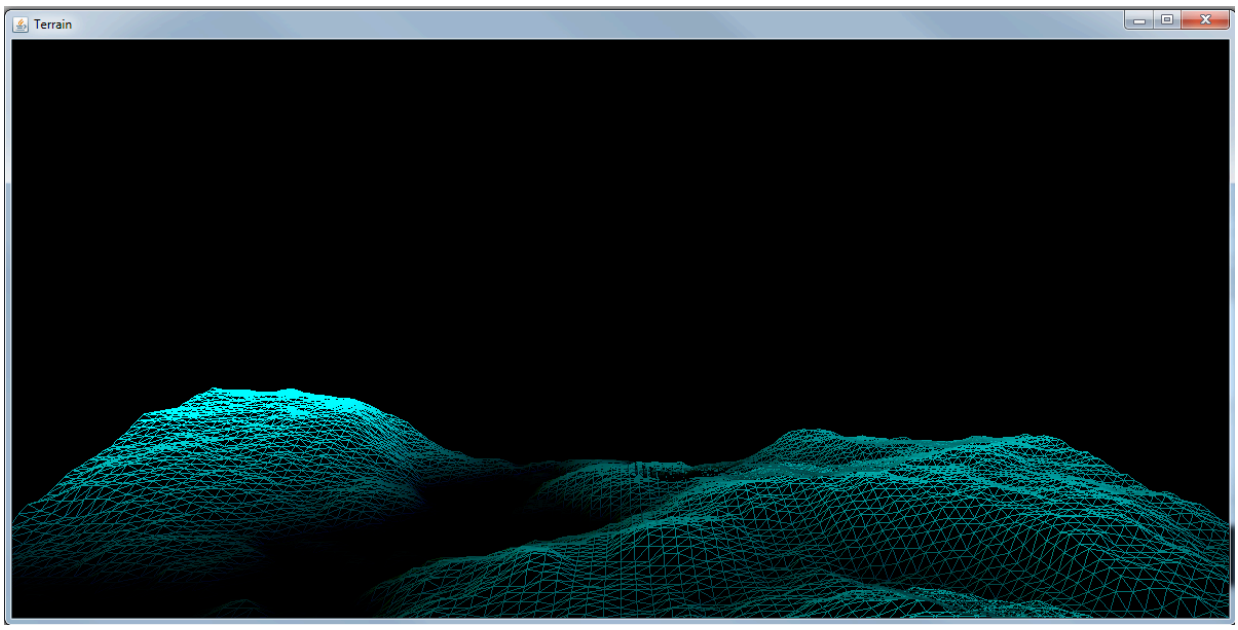

Abbildung 2 Beispielansicht der fertigen Landschaft

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Heinrich Hußmann
Henri Palleis

Aufgabe 3: Der Algorithmus von de Casteljau

Sie haben in der Vorlesung den Algorithmus von de Casteljau kennengelernt. Recherchieren Sie noch etwas weiter!

i. Beschreiben Sie den Algorithmus in eigenen Worten (100-200 Wörter)

ii. Implementieren Sie ein Programm in JavaScript, Java oder JOGL, das zu gegebenen Kontrollpunkten anhand des Algorithmus von de Casteljau eine Annäherung an eine Bézier-Kurve zeichnet

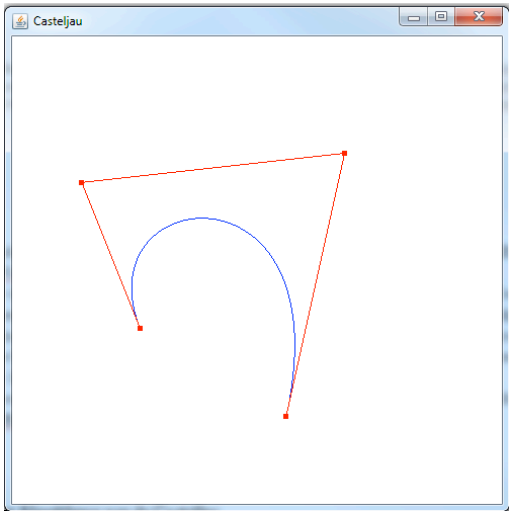

Abbildung 3 **Beispielausgabe für Aufgabe 2**

Viel Erfolg.