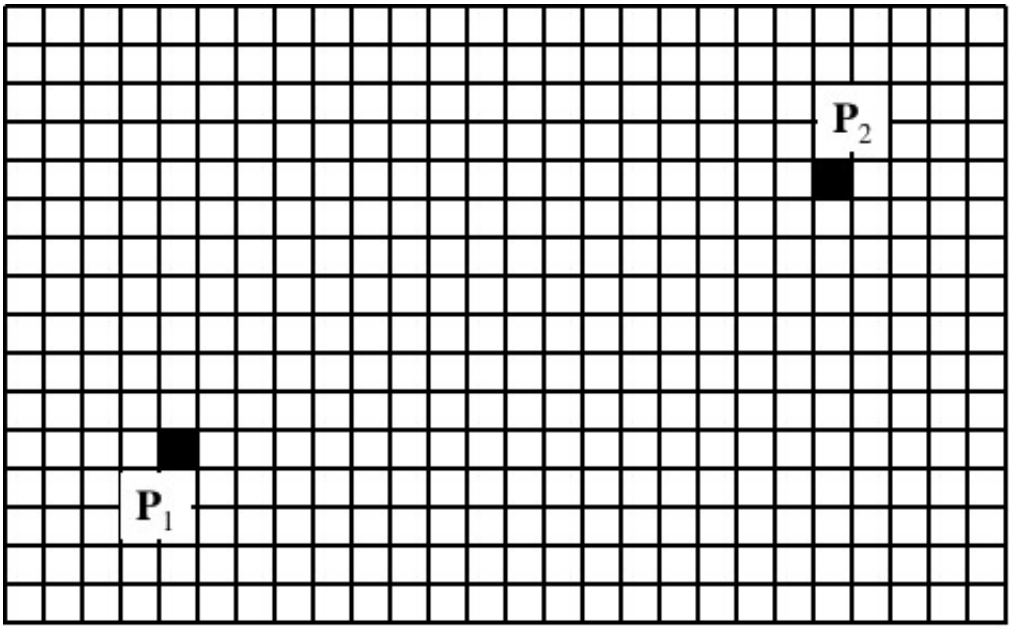
Übungsblatt 6

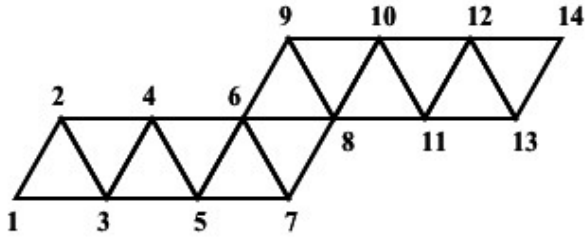
Aufgabe 1. Bestimmen Sie alle Pixel auf der Linie zwischen den Punkten $\mathbf{P}_1$ und $\mathbf{P}_2$ mit Hilfe des Algorithmus von Bresenham. Zeichnen Sie dabei auch alle benötigten Hilfselemente ein.



Aufgabe 2. Die nachfolgende Abbildung zeigt ein zu rasterisierendes Dreieck mit den Eckpunkten $\mathbf{a} = (3, 2, 3)^T$, $\mathbf{b} = (3, 6, 8)^T$, $\mathbf{c} = (9, 2, 15)^T$ sowie einigen schon berechneten Werten eines z-Buffers. Ergänzen Sie die fehlenden z-Werte.

0	0	0	0	0	0	0	0	0	0
0	0	3						15	0
0	0							0	0
0	0					0	0	0	0
0	7				0	0	0	0	0
0	0	8	0	0	0	0	0	0	0

Aufgabe 3. Die nachfolgende Abbildung zeigt 14 Punkte, die zusammen 12 Dreiecke bilden.



Schreiben Sie ein kleines OpenGL-Programm, welches die Geometrie der abgebildeten Dreiecksmenge mittels VBOs und die Topologie mittels folgender OpenGL Primitive darstellt:

1. `GL_TRIANGLES`. Achten Sie dabei auf die konsistente Orientierung der Dreiecke.
2. `GL_TRIANGLE_STRIP`. Hierbei soll die gesamte Dreiecksmenge durch *EIN* Triangle Strip dargestellt werden.

Aufgabe 4. Schreiben Sie ein einfaches OpenGL-Programm, zur Darstellung von 3D-Punktwolken. Beim Aufruf von

```
python oglViewer.py objectPoints
```

soll ihr Programm die Punkte aus der Datei `objectPoints` einlesen und eine orthographische Parallelprojektion der Punkte unverzerrt in einem GLUT-Fenster darstellen. Dabei sollen Änderungen der Fenstergröße berücksichtigt werden, so dass das eingelesene Modell immer unverzerrt angezeigt wird. Weiterhin soll es möglich sein, das Modell um den Mittelpunkt seiner Bounding-Box mit den Tasten

- `x`, `X` im bzw. gegen den Uhrzeigersinn um die `x`-Achse zu drehen.
- `y`, `Y` im bzw. gegen den Uhrzeigersinn um die `y`-Achse zu drehen.
- `z`, `Z` im bzw. gegen den Uhrzeigersinn um die `z`-Achse zu drehen.

Unter `http://www.mi.hs-rm.de/~schwan/Vorlesungen/GenCG` finden Sie – als *OpenGL-Einstiegsdroge* – das Python Skript `oglTemplate.py`.