

# Übung zu Computergraphik I

## – Übungsblatt 10 –

Lehrstuhl für Computergraphik  
und Multimediasysteme

Hendrik Hochstetter, John Rickard, Rene Winchenbach, Andreas Görlitz

**Abgabe:** Bis spätestens Donnerstag 25. Juni 2015, 10 Uhr

**Besprechung:** Mittwoch 1. Juli 2015 und Donnerstag 2. Juli 2015

**Hinweise:** Bearbeitungen bitte mit Name, Matrikelnummer und Übungsgruppe beschriften und zusammengeheftet in den Briefkasten vor Büro H-A 7115/1 werfen. Programmieraufgaben bitte per Mail mit Name und Matrikelnummer an Ihren jeweiligen Tutor senden. Geben Sie bei dabei nur Ihre modifizierte Quelltextdatei als Anhang ab.

### Aufgabe 1 Clipping von Strecken - Liang-Barsky-Algorithmus (1 Punkt)

Gegeben seien die folgenden zwei Punkte in Window-Koordinaten:

$$\mathbf{P}_1 = \begin{pmatrix} -2 \\ 3/4 \end{pmatrix}, \quad \mathbf{P}_2 = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$$

Im Folgenden soll ein Clipping der Strecke  $\overline{\mathbf{P}_1\mathbf{P}_2}$  gegen die erweiterten Fensterkanten des Bereichs  $[-1, 1]^2$  mit Hilfe des Liang-Barsky-Algorithmus durchgeführt werden.

- 1.1 Geben Sie den WEC-Vektor (Window-Edge-Coordinates) der beiden Punkte an.
- 1.2 Führen Sie ein Trivial-Accept bzw. Trivial-Reject durch, um zu prüfen, ob und mit welchen Fensterkanten ein Clipping durchgeführt werden muss.
- 1.3 Bestimmen Sie Anfangs- und Endpunkt  $\mathbf{Q}_1, \mathbf{Q}_2$  der aus dem Clipping resultierenden Strecke. Hinweis: Führen Sie eine Schnittpunktberechnung entsprechend des Ansatzes aus der Vorlesung durch.

### Aufgabe 2 Window-to-Viewport Mapping (1 Punkt)

Geben Sie eine Abbildung an, die Window-Koordinaten  $\begin{pmatrix} x \\ y \end{pmatrix}$  im Wertebereich  $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$  auf den Bildschirm mit Gerätekoordinaten  $[0, u_{\max}] \times [0, v_{\max}]$  abbildet. Die Abbildung soll das Höhen- und Seitenverhältnis beibehalten und den Ausgabebereich optimal ausnutzen. Stellen Sie eine  $3 \times 3$ -Matrix auf, die die Transformation beschreibt.

**Aufgabe 3 Sutherland-Hodgeman Algorithmus (1 Punkt)**

Gegeben sei ein Polygon in Normalized Device Coordinates, aufgespannt durch die Punkte

$$\mathbf{P}_1 = \begin{pmatrix} -\frac{3}{2} \\ 1 \end{pmatrix}, \quad \mathbf{P}_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \quad \text{und} \quad \mathbf{P}_3 = \begin{pmatrix} \frac{1}{2} \\ -\frac{3}{2} \end{pmatrix}.$$

- 3.1 Welche Voraussetzung muss für ein Clipping-Polygon gegeben sein, damit der Sutherland-Hodgeman Algorithmus funktioniert? Begründen Sie Ihre Aussage.
- 3.2 Berechnen und skizzieren Sie die einzelnen Schritte des Sutherland-Hodgeman Algorithmus' für das Clipping gegen  $[-1, 1]^2$ .

**Aufgabe 4 Linienrasterisierung - Bresenham-Algorithmus (1 Punkt)**

Nehmen Sie als Ausgangsbasis für die folgende Aufgabe das auf unserer Website bereitgestellte Programmgerüst `ueb10.zip`. Beachten Sie die technischen Hinweise auf der Übungsseite. Das Programm ermöglicht in der vorliegenden Form das Zeichnen von Linien, welche die Voraussetzung für den Rasterisierungsalgorithmus von Bresenham erfüllen. Hierzu wird durch zwei aufeinanderfolgende Mausklicks im Zeichenbereich je eine Strecke definiert.

Falls die Voraussetzung für den Bresenham-Algorithmus nicht erfüllt ist, wird ein Fenster mit einer entsprechenden Warnung angezeigt. Erweitern Sie das Programm, so dass beliebige Linien gezeichnet werden können.

**Hinweis:** Führen Sie eine Umwandlung der Endpunktkoordinaten genau dann durch, wenn eine Fehlermeldung ausgegeben würde. Überlegen Sie für jeden Fall, welche Veränderung durchzuführen ist, damit die Voraussetzung für den Bresenham-Algorithmus wiederhergestellt ist. Achten Sie außerdem darauf, dass bei der Ausgabe in den `vertices`-Buffer die Raster-Punkte an korrekter Stelle ausgegeben werden.