Kapitel 5

Clipping

Ziel: Nur den Teil eines Objekts darstellen, der innerhalb eines Fensters sichtbar ist.

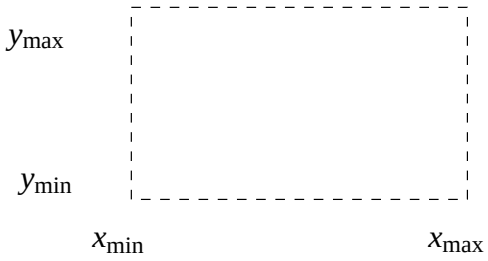

Abbildung 5.1: Clip-Fenster

5.1 Clipping von Linien

Zu einer Menge von Linien sind jeweils neue Anfangs- und Endpunkte zu bestimmen, welche komplett im Clip-Fenster liegen.

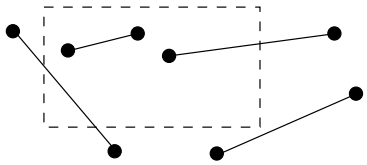

Abbildung 5.2: Ausgangslage beim Linien-Clipping

Idee von Cohen & Sutherland

Teile Ebene anhand des Clip-Fensters in 9 Bereiche ein, beschrieben durch 4-Bit-Bereichscode:

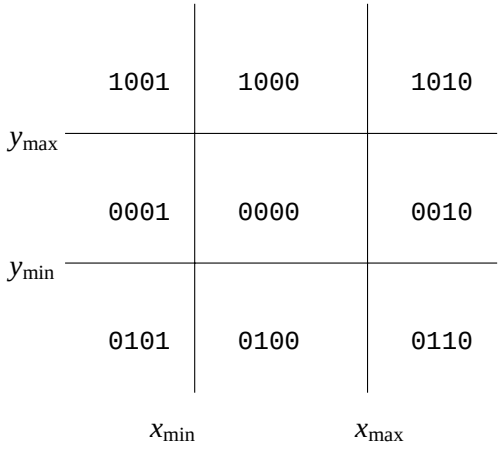

Abbildung 5.3: Bereichscodes nach Cohen & Sutherland

Bit 0: links vom Fenster
Bit 1: rechts vom Fenster
Bit 2: unter dem Fenster
Bit 3: über dem Fenster

Sei $\overline{P_1P_2}$ eine Linie. Dann gilt bei einer bitweisen Verknüpfung:

code (P_1) AND *code* $(P_2) \neq 0 \Rightarrow \overline{P_1P_2}$ komplett außerhalb (beide Punkte auf derselben Seite)

code (P_1) OR *code* $(P_2) = 0 \Rightarrow \overline{P_1P_2}$ komplett innerhalb (beide Punkte im Clip-Fenster)

In den anderen Fällen wird $\overline{P_1P_2}$ mit einer Fensterkante geschnitten und der Test mit der verkürzten Linie erneut ausgeführt.

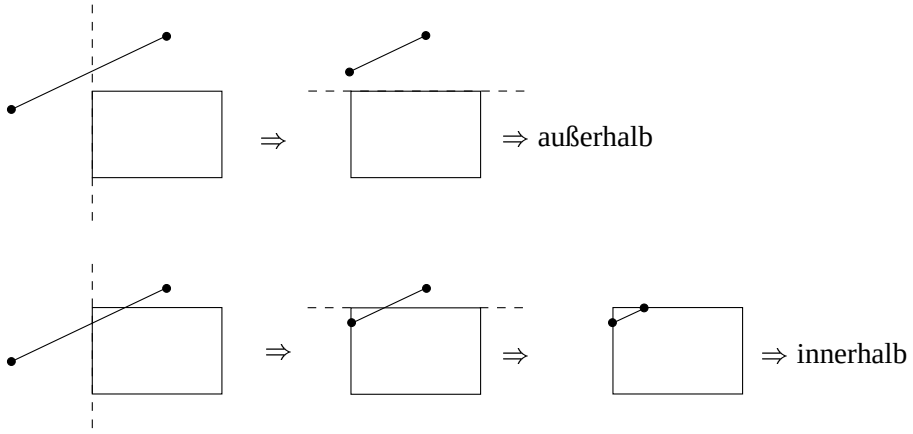

Abbildung 5.4: Möglichkeiten bei zwei außerhalb liegenden Punkten

```
private static final byte CENTER  = 0;
private static final byte LEFT    = 1;
private static final byte RIGHT   = 2;          // 4-Bit-Bereichscodes
private static final byte BOTTOM  = 4;
private static final byte TOP     = 8;

/**
 *  Liefert den region code fuer den uebergebenen Punkt
 *
 *  @param x Die x-Koordinate des zu testenden Punktes
 *  @param y Die y-Koordinate des zu testenden Punktes
 *
 *  @return Der region code in einem Byte kodiert
 */
public byte region_code(int x, int y) {
  byte c = CENTER;                              // zunaechst: im Zentrum

  if (x < xmin)                                 // falls links vom Fenster
    c = LEFT;                                   // links vermerken
  else                                          // falls nicht links
    if (x > xmax)                               // falls rechts vom Fenster
      c = RIGHT;                                // rechts vermerken

  if (y < ymin)                                 // falls oberhalb
    c |= TOP;                                   // oberhalb hinzufuegen
  else                                          // falls nicht oberhalb
    if (y > ymax)                               // falls unterhalb
      c |= BOTTOM;                              // unterhalb hinzufuegen

  return c;                                     // region code zurueck
}
```

5.2 Clipping von Polygonen

Für das Clippen von Polygonen reicht es nicht, jede beteiligte Kante zu clippen:

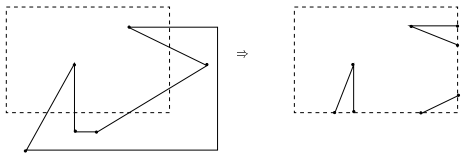

Abbildung 5.5: Zerfall eines Polygons bei reinem Linien-Clippen

Vielmehr müssen zusätzlich die Ein- und Austrittspunkte verbunden werden, um nach dem Clipping wieder ein Polygon zu erhalten:

Obacht: Bzgl. der Ecken des Clip-Windows ist eine Spezialbehandlung erforderlich:

Abbildung 5.6: Verbinden der Ein- und Austrittspunkte

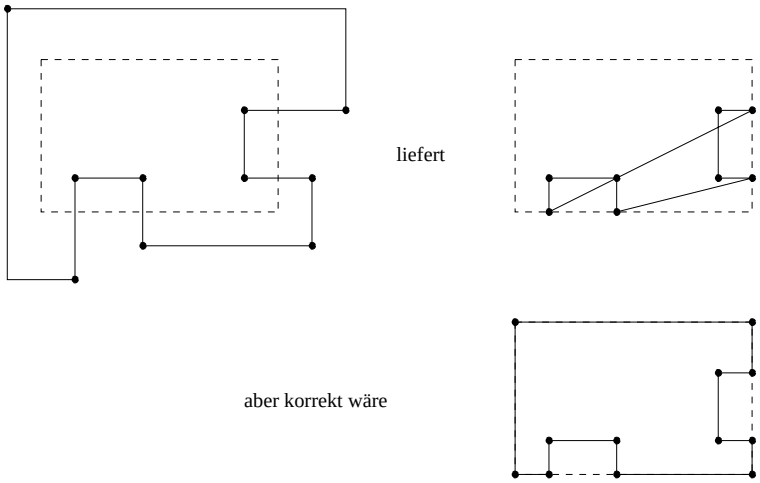

Abbildung 5.7: Spezialbehandlung an den Clipfensterecken

Der **Sutherland & Hodgman-Algorithmus** clippt an vier Fensterkanten nacheinander:

```
fuer jede Clipping-Gerade E tue:
  fuer jeden Polygonpunkt P tue:
    falls P sichtbar: uebernimm ihn
    falls Kante von P zu seinem Nachfolger E schneidet:
          uebernimm den Schnittpunkt
```

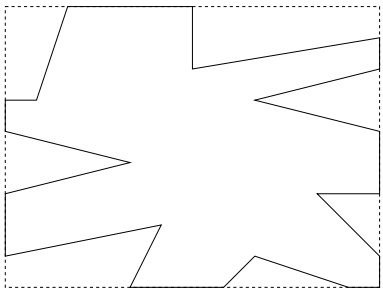

Abbildung 5.8: Vom Sutherland-Hodgman-Algorithmus geclipptes Polygon

5.3 Beispiel-Applet zu 2D-Operationen

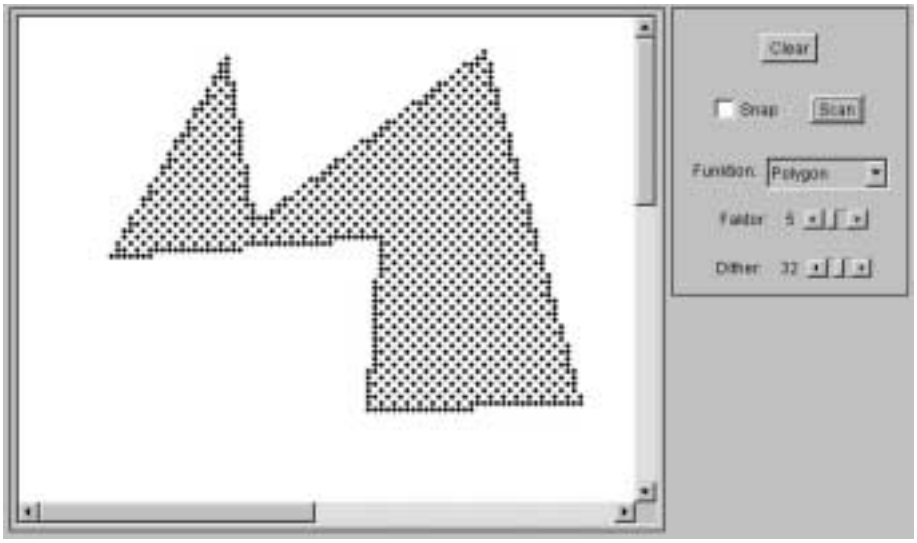

Abbildung 5.9: Screenshot vom 2D-Basic-Applet