

Übung Nr. 2:

Im Konsole-Fenster sollen geladene dreidimensionale grafische Flächenmodelle mit ASCII-Zeichen dargestellt werden (Menüpunkt 'w' des Programms). Als Symbol zur Flächenfüllung soll jeweils die Ziffer dienen, die den Index der jeweiligen Fläche angibt (Abb.1). Dazu ist unter <http://homepages.thm.de/christ/> das vorbereitete MS-VC-Projekt „WireCullFill“ zu erweitern. Die gleichnamige C-Datei (WireCullFill.c) ist nicht komplett, sie braucht aber nicht vervollständigt zu werden; denn die fehlenden Anweisungen ihrer Funktionen (Gegenstand der Übungen zur BSc-Vorlesung) sind vorcompiliert in der Datei WireCullFill.obj enthalten, die anstelle der Quelle im Projekt eingebunden ist. Auf ähnliche Weise sind auch die übrigen Projekt-Dateien dem Projekt hinzugefügt worden.

Zur Auswahl des benötigten Füllalgorithmus kann vorausgesetzt werden, daß die dargestellten Objekte ausschließlich von konvexen Flächen umschlossen seien, d.h., jedes Bild- bzw. Fensterzeilen-Fragment („Span“), das zur Füllung einer Fläche beiträgt, sei zusammenhängend (es wird nicht durch Symbole anderer Flächen unterbrochen).

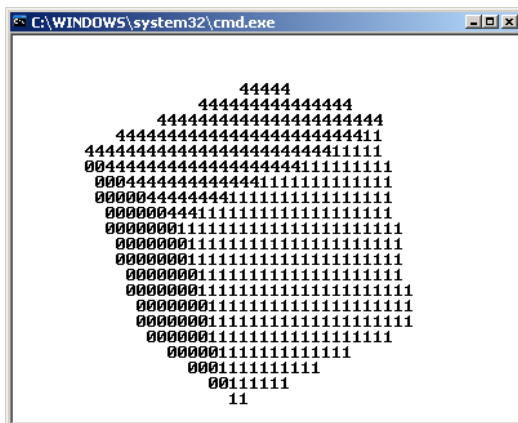


Abb. 1: WireCullFill.exe

Die Lösung dieser Aufgabe verlangt knapp 15 Zeilen Code (s. Bezeichner MORE_FILL) und bezieht die folgenden C-Dateien und -Funktionen in die Lösung ein:

fillOps.c	Die Header-Datei (fillOps.h) ist fertig; sie braucht nicht verändert zu werden.
fillOps.h	Die C-Datei enthält die drei kurzen Funktionen, die zur Lösung dieser Aufgabe ausreichen:
	<code>void initF4F (int *endL, int *endR)</code>
	ist bereits fertig. Sie initialisiert die beiden Felder endL[] und endR[] (deklariert in fillOps.h); diese sollen später für jede Zeile, die eine Fläche belegt, die Koordinaten (Pixel-Nr.) des linken und des rechten Rands aufnehmen. (initF4F() wird in conDrawCGFobj() aufgerufen, sie ist bereits dort eingesetzt.)

Die (unvollständige) Funktion

```
void markF4F (int x, int y, int *endL, int *endR)
```

soll den Initialisierungswert ersetzen, indem sie (während der Ziehung der Begrenzungslinien einer Fläche mit `MidpointLine()`) „notiert“, auf welchen Fensterzeilen, jeweils vom linken bis zum rechten Span-Ende, die gewünschten Farbpixel (hier: Füllsymbole) zu setzen sind.

Erwartungsgemäß mußte zur Span-Markierung der Bresenham-Algorithmus um den Aufruf von `markF4F()` erweitert („aufgebohrt“) werden. Das ist in der hier eingebundenen, vorcompilierten Fassung bereits geschehen.

Mit

```
void fillF (int *endL, int *endR, char symbol)
```

sollen schließlich die Pixel (hier: Symbole) zur Flächenfüllung zeilenweise von unten nach oben gesetzt werden.

(Die Zuweisung des jeweiligen Symbols an jede Fläche ist beim Laden jedes Objektes erfolgt; sie war somit Gegenstand früherer BSc-Übungen.)