Übung Nr. 2:

Mit ASCII-Zeichen ist im Konsole-Fenster als Grafik eine Analog-Uhr zu erstellen (Abb.1). Unter http://homepages.thm.de/christ/ kann das hierzu benötigte Projekt von MS-VC heruntergeladen werden. (Das heruntergeladene Projekt läßt sich compilieren, es tut aber vorerst nichts Sinnvolles.)

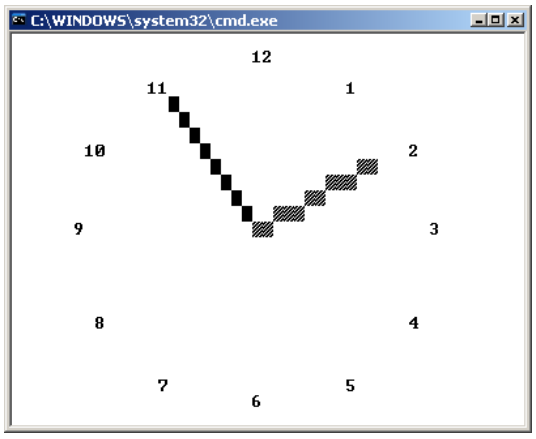

Abb. 1 `BresenClock.exe`

Zwei Entwicklungsstufen sind vorgesehen, die als ausführbare Programme im o.a. Projekt enthalten sind (`BresenClock(1).exe` und `BresenClock(2).exe`); ihre Erstellung in dieser Reihenfolge ist nicht zwingend, aber ratsam.

Wieder eingesetzt werden die aus der 1. Übung bekannten Dateien `conDraw.*` und `lineOps.*`; erstere werden weiterhin bereitgestellt, letztere wurden für alle Eventualitäten in compilierter Form dem Projekt beigefügt (*.obj-Dateien im Src- u./o. _Lib-Verzeichnis).

Folgende Vorgehensweise ist besonders zu empfehlen:

1 Matrizen-Operationen

Zunächst sollte die Datei `MatrOps.c` behandelt werden, denn sie enthält die zwei mathematischen Operationen, die zur Darstellung der Uhr unentbehrlich werden: Die Funktion `int Identity (float *matrix, int dim)` belegt eine Matrix mit den Werten einer Einheitsmatrix; sie ist „gebrauchsfertig“. Demgegenüber muß die vorbereitete Funktion `int matMul (float f1, float *mi1, int z1, int s1z2, float f2, float *mi2, int s2, float *mout)` erst codiert werden. Sie ermöglicht die Multiplikation zweier Matrizen beliebiger Größe. (Die skalaren Faktoren `f1` und `f2` sind aus mathematischer Sicht willkürlich; sie sollen lediglich Anwendungen erleichtern, bei denen alle Elemente der jeweiligen Matrix mit einem konstanten Faktor zu multiplizieren sind.) Die übrigen Funktionen in dieser Datei werden erst bei späteren Übungen benötigt.

Matrizen sind im Rahmen dieser Aufgabe (praxisnah) als eindimensionale Felder gespeichert. Die zwei Funktionen in dieser Datei sind unabhängig von der Größe der verwendeten Matrizen. Wo innerhalb dieses Projektes eine feste Matrix-Größe erwartet wird, sind 4x4-Matrizen vorgesehen.

Nach korrekter Codierung sollte das compilierte Programm den Stunden- und den Minuten-Zeiger lediglich in der 12-Uhr-Stellung (im Stillstand) anzeigen. Um Funktionsstörungen auszuschließen, sollte zuvor darauf geachtet werden, daß die Präprozessor-Anweisung `#define MORE_MATR1` am Anfang von `MatrOps.h` wirksam (nicht auskommentiert) ist.

2 Grafik-Operationen

In der Datei `GrafOps.c` sind drei Funktionen vorbereitet:

```
int scale     (float sx, float sy, float sz, float *posMat)
int translate (float xt, float yt, float zt, float *posMat)
int rotate    (float theta, int axis, float *posMat)
```

Hier brauchen nur die Werte zugewiesen zu werden, die die Skalierungs-, Translations- und Rotationsmatrix von einer Einheitsmatrix unterscheiden, damit die jeweilige Matrix mit der Positionierungsmatrix der Objekte (hier: der beiden Uhrzeiger) multipliziert werden kann.

Auch hier handelt es sich um C-Funktionen für 4x4-Matrizen (was eine Vorarbeit für die spätere Anwendung auf dreidimensionale Grafiken darstellt). Noch vorhandene Unklarheiten werden durch Hilfestellung in den betreuten Übungsblöcken beseitigt.

Die erfolgreiche Codierung dieses Übungsteils sollte die Funktionsweise von `BresenClock(1).exe` aufweisen. Hier ist wieder darauf zu achten, daß in `GrafOps.c` die Zeile `#define MORE_GRAPH1` aktiviert (nicht auskommentiert) ist. Auch in dieser Datei gibt es Funktionen, die (in Zusammenhang mit `MORE_GRAPH2`) erst später benötigt werden.

Es lohnt sich, während und nach Erledigung dieser relativ einfachen Schritte sich mit der Arbeitsweise des gesamten Programms auseinanderzusetzen; sehr zu empfehlen ist der Einsatz des Debuggers. Typisch für die Arbeit mit Bildern und Grafiken ist dabei, daß recht viel (hier z.T. bereitgestellter) Code und Geduld nötig sind, bevor ein Bild zu sehen ist; zudem ist oft (wie hier) das Bild nicht während seiner Entstehung, sondern erst nach seiner Fertigstellung sichtbar.

3 Animation-Feineinstellung und Fertigstellung

Die Datei `BresenClock.c` enthält vor allem die Funktionen, die notwendig sind, um im Konsole-Fenster zu zeichnen (zu erkennen an Namen, die mit „con“ beginnen). Nachdem diese (an sich eher lästige) Arbeit erledigt ist, wäre interessant, sich mit der Wirkungsweise der einzelnen Funktionen vertraut zu machen, weil sie eine Miniatur dessen darstellen, was große, elaborierte Grafik-Systeme tun.

Hier ist nachzuvollziehen, wo und wie die Animation der Uhrzeiger erfolgt, damit die Positionierung des Stundenzeigers kontinuierlich an jene des Minutenzeigers angepaßt wird und nicht mehr so sprunghaft zwischen den Stunden wechselt (`BresenClock(2).exe` – zuvor `#define MORE_BRESEN` in `BresenClock.c` aktivieren!).

Vor Präsentation und Abgabe sollte in dieser Datei (in `main()`) dafür gesorgt werden, daß sich das Programm mit dem werten Namen und der Matrikel-Nummer seiner Autor/inn/en meldet.