

Hinweise zum Grafikmodus „Betriebssystem-Entwicklung“

1. Textausgaben

Allgemeines

Die Vorgabe enthält Schriftarten sowie Funktionen, um damit rudimentär Text im Grafikmodus ausgeben zu können. Es handelt sich um nicht-proportionale Schriften, bei denen alle Zeichen die gleiche Breite haben. Die Schriftarten sind als monochrome Rastergrafiken gespeichert und die Rastergrafik für jedes Zeichen hat neben der gleichen Breite auch die gleiche Höhe. Dies vereinfacht die Handhabung sehr stark, im Vergleich zu True-Type-Schriftarten, bietet aber im Gegenzug weniger Möglichkeiten für die Darstellung.

Schriftarten

Für jede Schriftart gibt es eine Header-Datei (`devices/fonts/Font_XXX.h`) welche die Rastergrafiken für alle Zeichen beinhaltet. Die Schriften wurden ursprünglich mit dem Werkzeug `cp12fnt` erzeugt, welches noch vom Amiga-Betriebssystem (Amiga ist ein Home-Computer aus den 80er Jahren) stammt und welches im Netz nicht mehr auffindbar ist. Die Schriften wurden teilweise auch in Linux von X11 verwendet und es finden sich weitere solche Schriften im Internet, die man in sein hhuTOS auch integrieren kann.

Die Klasse `Fonts` (von Jakob Falke) in `devices/fonts/Fonts.h` definiert mithilfe von Templates C++ Objekte für die vorhandenen Schriftarten, was Verwendung erleichtert. Die Funktion `getChar` liefert einen Zeiger auf die Rastergrafik für ein Zeichen, welches per ASCII-Code ausgewählt wird.

Ausgabefunktionen

Für die Ausgabe von Zeichenketten mit einem bestimmten Font in einer gegebenen Farbe gibt es in der Klasse `VGA` die beiden Methoden `drawString` und `drawMonoBitmap`. Die Methode `drawMonoBitmap` ist eine Hilfsfunktion für `drawString` und gibt die Rastergrafik eines Zeichens an einer gegebenen Position mit der gewünschten Farbe aus (mithilfe von `drawPixel`).

Hinweis: eine vergleichsweise einfache Möglichkeit, um True-Type-Fonts als Rastergrafiken in hhuTOS nutzen zu können geht über das BDF-Format (Glyph Bitmap Distribution Format von Adobe), wofür es kostenlose Werkzeuge im Internet gibt. Eine echte Unterstützung von True-Type-Fonts bei Zeichen dynamisch berechnet und dargestellt werden ist sehr komplex.

2. Einbinden und Ausgaben von Bitmaps

In der Vorgabe befindet sich das Logo der HHU als Bitmap in der Datei `bmp_hhu.cc`. Das Format der Datei ist unkomprimiert und pro Pixel werden 4 Bytes gespeichert, je 1 Byte für Rot, Grün und Blau und 1 Byte für den Alpha-Wert (0 - 255 Deckkraft).

In GIMP kann eine Grafik als C-Sourcecode (*.c) exportiert werden. In den Optionen beim Export sollten **keine** Glib Typen verwendet werden und das Bild sollte **mit** Alpha-Kanal exportiert werden. Außerdem darf keine Kompression aktiviert werden. Im Header der Datei steht als Name für die struct `"gimp_image"`, was man in einen geeigneten Namen abändern sollte.