EI Info3 2010-11	*INFORMATIK* Stunde vom 27.09.2010	110010010101 100010100101 010101010101 101010101010 101010101010

Wir haben in dieser Stunde noch einmal die Idee der Codierung von Daten mittels Bits (also Zuständen von 0 oder 1) wiederholt. Im Anschluss habt ihr überprüft, dass beispielsweise Farben so codiert werden. Als Projekt habt ihr dann die Aufgabe erhalten, eine einfache Grafik zu codieren.

Farbcodierung

Zuerst einmal zur Farbkodierung. Man kann alle Farben aus wenigen Grundfarben „mischen". Das wisst ihr ggf. aus der Physik oder aus der Kunst. In Druckern sind es häufig die vier Farben Cyan, Magenta, Yellow und Black (also ein Blau-, Pink, Gelbton und Schwarz):

An eurem Bildschirm wird jeder Pixel (ein Bildpunkt, es gibt an meinem Rechner 1024x600 davon, also grob 600.000) durch 3 Minikristalle gebaut, die in RGB leuchten (also rot, grün bzw. blau). Damit kann auch hier jedem Pixel eine Farbe zugewiesen werden. Wenn ihr mit der Lupe dran geht, erkennt ihr diese Struktur!

Auch in den geläufigen Programmen wie im Internet (html, programmieren wir auch noch in der Kursstufe) ist eine Farbe RGB-codiert. Es werden Anteile dieser Farben zusammengemischt. Dabei kann man je Farbton 256 Abstufungen wählen. Dann wird klar, dass ein Farbton gerade durch ein Byte codiert wird, weil das ja 256 verschiedene Zustände kodieren kann. Ein Wunschrot könnte dann bspw. so aussehen: 10001001. Dazu noch ein Wunschgrün bzw. –blau und fertig ist die Farbmischung. Diese lange Zeichenkette wäre allerdings umständlich zu codieren. Im html-Code ist ein Befehl hierzu: bgcolor=#FARBTON, damit wird die Hintergrundfarbe einer html-Seite festgelegt. Hier müsste man jetzt bspw. folgendes eintippen: „bgcolor=#100010011000100110001001" und das nervt. Daher wird auch hier hexadezimal codiert, denn im Hexadezimalsystem lässt sich ein Byte mit nur 2 Ziffern darstellen! Dann schreibt sich eine Farbe bspw. einfach als F1012A. Das ist viel kürzer. Hier sieht man einmal mehr, dass das Hexadezimalsystem in der direkten Programmierung einfach nützlicher ist. Maschinenintern muss natürlich wieder umgerechnet werden, aber das geht sehr sehr schnell. *Noch ein kleiner Hinweis: die Raute (#) im Code ist ein Platzhalter dafür, dass der PC weiß, dass jetzt eine Grafik der Form (RotRot)(GrünGrün)(BlauBlau) kommt. Diese Platzhalter sind wirklich wichtig, damit der PC nicht durcheinander kommt. Das werden wir bei unserer Grafikcodierung wiederfinden!*

Grafikcodierung

Wie wird nun im PC eine Grafik codiert? Wir können schon alle Texte plus Formate einspeisen, aber Grafiken sind ja zweidimensional, anders als einfache Textketten. Der PC lässt sich für seine Pixel auf den Bildschirm jeweils ein Signal geben und schaltet entsprechend um. Da wir es uns einfacher machen wollten, haben wir eine reine Zweifarbgrafik codiert. Das wäre bspw. hell/dunkel, oder blau/gelb oder was auch immer. Muss man nur einmal festlegen. Der Vorteil davon ist, dass der Code viel einfacher wird. Eine 1 ist bei mir gelb, eine 0 blau. Ich kann nun jede Pixelzeile der Grafik in 10-Ketten codieren, bspw.:

110010010101
100010100101
010101010101
101010101010
101010101010

Da ich aber dem PC eine ZeichenKETTE einspeise, weiß der jetzt nicht, wann die nächste Zeile beginnt. Auch weiß er nicht, wie groß die Grafik ist. Dies ist wichtig, da immer Speicher zugewiesen werden muss und der sollte ja auch ausreichend sein. So wird unser Grafikformat erst einmal mit einer Codierung beginnen, die sagt, wie groß die Grafik eigentlich ist. Wir lassen nur ein Format von 1…16 mal 1…16 Pixel zu.

Das steht am Anfang: 10110100 wären zum Beispiel 12x5 Pixel (siehe Bild oben). Nach 8 Zeichen der eingelesenen Zeichenkette

10110100 110010010101 100010100101 010101010101 101010101010 101010101010

beginnt für den PC das Bild (die Leerzeichen habe ich gesetzt, damit klarer ist, was passiert. Hätten wir nicht nur blau-gelb, müssten wir ggf. hier noch sagen, was 0 bzw. was 1 bedeuten. Dann wäre unsere Zeichenkette eben länger. Nun folgt für den PC eine Schleife und das solange, bis er auf 12 hochgezählt hat; er nimmt die Farbe 1 bzw. 0 auf und setzt sie ins Blankobild ein. Danach springt er in die nächste Bildzeile und liest die Zeichenkette weiter. So sind Grafikformate aufgebaut. Man könnte am Ende unseres Formates noch eine besondere Zeichenfolge machen, um zu sagen: Grafik beendet. Es sind viele weitere Ideen möglich!